SD MESXES

Número 14
Mayo 2000
550 pta

ESTOESLOKEHAYRIO

SD MESXES #14 - MAYO 1900 (perdón, 2000) - 550 pta / 3 uros

Pues eso, que esta vez esto es lo que hay:

Edita: CLUB MESXES

(cuándo el tiempo y la autoridad lo permiten)

Redacción

Ramón Serna *(a título póstumo)*

Néstor Soriano *(cuando no está en Barcelona)*

Marcos Rosales *(cuando se acuerda de que existimos)*

SaveR *(cuando se lo permite la Jolís)*

Maquetación

Los arriba firmantes, en las condiciones expuestas

Colaboran

Marcelino Luna *(este hombre tiene el cielo ganado)*

Juan José Luna *(sin ser tan buena persona, se hace bastante cargo)*

Funet, Iván Isidro, The Matra Corporation

Club Mesxes quiere pedir disculpas a sus suscriptores y simpatizantes por los descomunales retrasos en la publicación del fanzinne y por el escaso grosor de este número en concreto, pero no nos queda tiempo ni para eso.

E-mail: ramoni@mallorcaweb.net
konamiman@geocities.com
savermsx@yahoo.com

Web: http://mesxes.msx.tni.nl
C/C Caja Postal: 29-91.392.819
Nuevo capitán: Konami Man

イントロデュクズィオン*

**N. del T.:* いんとろっでゅくずぃおん

Sectarios todos, he de daros una noticia cuya significancia va más allá de lo inimaginable. El hasta ahora capitán del buque insignia Secta Mesxes I, el gran Ramonijo Ijo, acaba de abdicar, traspasando todos sus poderes a la persona del antaño redactor jefe Konami Man (lo mesmo), debido sin duda a los achaques de la avanzada edad que no tiene, lo cual le impedía desempeñar sus funciones de la forma más apropiada para el bien de la comunidad. O eso, o que ya estaba hasta los cojines del campotraviesa, los retrasos, el pajas maker, el ventanucos y el estress que todo ello conllevaba, que debía además ser compaginado con el estress propio de la vida estudiantil. La pregunta es: si todo un capitán como el Capitán Moncho ha sucumbido a las circunstancias colindantes, ¿cuánto tiempo aguantará el no menos grande pero igualmente humano y muy ocupado Konami Man, y por ende el fanzine? Esperemos que mucho, pero para ello es necesaria vuestra ayuda: mandadnos artículos, trukis, anuncios, mandadnos a la mierda si quereis, ¡¡pero mandadnos algo!!



El momento del traspaso de poderes, inmortalizado.

Y hablando de campotraviesa, nos hallamos en estos momentos inmersos en ella a un nivel pocas veces alcanzado. ¿Sus imaginais que la próxima ru (para la cual pretendemos tener listo el fanzain) es el 30 de abril, lo parto hacia basuralona el domingo 23 a las 22:30, y ahora son las 6:00 del sábado 22 (en los últimos coletazos de la noche de la decadencia) y aún estamos redactando cosas? Pues eso. Parece mentira pero no hemos podido (???) hacer las cosas en todo el tiempo pasado desde el SD#13. Conclusión: SD#14 canijo, mal hecho (esperemos que no mucho) y sectarios destrozados, como siempre (manque unos aguantamos más que otros, también como siempre). Pero esperemos que os hagais el suficiente cargo.

Eso pues. Pasa la página y no nos pegues muy fuerte, porfa...

¡Alto! ¡No piense más! Tengo lo que necesita. Imagínese: todo lo que Vd. siempre deseó puede ser suyo ahora sin esfuerzo alguno. ¿Quién iba a decirle que algún día programaría Vd. el COMPASS? ¿O que sería aclamado mundialmente por haber ideado el BMX SIMULATOR para MSX2? ¿Y qué decir de la fama que conseguiría al imaginar un sistema de autoabsorción de direcciones de retorno de subrutinas de gestión de interrupciones, utilizando para ello la tabla de jerarquía de acceso hexadecimal al uso? Nunca lo habría conseguido de la forma con la que puede hacerlo ahora: el NéstorPlagio. El **NéstorPlagio** se compone de una suerte de colectividades, organizadas en palabras de 16 bits sin signo, con las que acceder al formato de la TPA es inusualmente sencillo y gratificante, al tiempo que permite vivir una experiencia completamente nueva. Una pequeña muestra de lo que puede conseguir:

- *Programar el bus de datos.*

- *Organización de ficheros pseudo-aleatorios.*

- *Metodología unificada (en conjunción con el NéstorMétodo).*

- *Compatibilizar cualquier ordenador con un TurboR previa inserción de un cartucho.*

- *El Chicharro Gordo.*

Cómo no, todo esto se lo entrego de corazón, no le costará ni un duro, y para hacer frente a su razonable desconfianza sepa Vd. que le ofreco también mi Garantía Total, basada en:

- *Las pruebas realizadas en BASIC son concluyentes.*

- *...no sabría decirle con exactitud...*

Todo ello acompañado de las instrucciones con la información técnica más completa para que Vd. desarrolle las futuras aplicaciones que desee.

¿Va a esperar más tiempo sabiendo además que recibirá de regalo una grabación del gotear de un grifo en una noche?


Antes



Después

Ya está quí la versión MSX de BEAT MANIA...

Tataku Mania

GAME

PRACTICE

OPTIONS

MSX SOUND SIMULATION

Presentación el 30 de Abril en la 17 Ru de MSX en Barcelona. Consiguelo ya en el CLUB MESXES.

MoonSound

NestorTRUCOS para NestorBASIC

A la pá de dió, mis queridos sectarios programantes. Sed bienvenidos a un artículo de ensamblador ciertamente atípico pero -espero- tan útil y ameno como los anteriores. Seguramente sus preguntaréis que lo cualo qué pinta un monólogo sobre NestorBASIC en un artículo que deberiese versar única y etsxclusivamente sobre nemónicos ensambladeros... la respuesta se hace cargo por sí misma, y se basa en tres raçaos:

1) Gran parte de los trukis que voy a etsxplicar se refieren al uso de rutinas ensambladeras en NestorBASIC, las cualas reducen enormemente el código BASIC necesario y por ende la memoria IDEM consumida, amén de proporcionar una velocidad inusitada tal que bla, bla...

2) Aprovecho para la presentación en sociedad (o mejor dicho, en secta) del NestorPreTer, mi nuevo pre-intérprete de BASIC, de lo cual hay autorrealización de cargo por medio de sendos ejemplos.

3) Mi ego tiene que vivir de algo.

Y sin más (more) pasamos ya al articulamen, ke la página de fanzain está por los cumulonimbos (croketamente a 2,5 pelas + IVA la página).

NestorPreTer: QUE LO CUALO QUE ES ELLO

Como no es muy difícil deducir por su nombre, es una utilidad developada por el menda. Se trata de un pre-intérprete de BASIC. Explicome: tú coges y vas y haces tu programa en el TED o en cualquier otro editor de textículos al uso, lo pasas por el NestorPreTer, y ya lo tienes convertido a un fichero BASIC de toda la vida que puedes asesinar, digo, ejecutar (creo que este chiste ya lo he, pero bué). ¿Y dónde está la gracia con respecto a facerlo directamente en BASIC "como antes"? Pues enresulta que en el pograma "fuente" hecho con el editor de textos puedes...

- Olvidarte de los números de línea, y usar etiquetas de línea en su lugar. Useáse en vez de GOTO 65000, ya puedes idemar GOTO ~A_LA_MIERDA.

- Incluir todos los comentarios que quieras, ya que serán ignorados, useáse no incluídos en el fichero "ejecutable" resultante.

- Usar macros para nombrar constantes, variables, cachos de código, etc. Por ejemplen, ínate LOCATE @FILA, @COLUMNA siendo sustituído por LOCATE X,Y.

...con lo cual te quedan listados más limpios, bonitos y legibles (oh). Omola, ¿veridez? En el **listado 1** tienes un idem de ejemplo que



simplemente carga NestorBASIC (hay que empezar por algo sencillo, ¿no?). Observa que creamos una matriz D donde metemos todas las variables al uso, no te acongojes, kel porké desto lo ekxplicaremos con detalle más adelante.

Como viene siendo habitualizable, NestorPreTer es gratuito y te lo puedes bajar de konamiman.msx.tni.nl, o pedírnoslo si estás tan rematadamente obsoleto que bla, bla... deprisita, que a partir de ahorina todos los ejemplos de algos BASIC que aparezcan en esta revista (¡fanzine!) estarán en formato NestorPreTer (al menos los que haga yo), y voy a suponer que estás famliarizado con su sintaxis/croquetas (uró). Si quieres más ejemplos de los aquí aparecidos, pídemelos también (téngolos).

NestorTRUKI el #0: CONSIDERACIONES GENERALES

Uró, qué miedo, qué título tan agresivo... casi que estoy por cambiarlo... pero paso.

Inate que eres un programador de ensamblador (que supongo que lo eres) y un alguien te encarga facer un editor de algos (sean mapas, espraits o firburcios nórdicos). Para un caso asín lo más práctico es usar el BASIC, y es entonces cuando piensas "podría usar NestorBASIC, pero, ¿pa qué? ¿Pa cagal.la?". Y entonces aparezco lo dispuesto a sacarte de tu ignoranciez, a base de patadas en los dientes si hace falta.



Aborejam, el problema secular de los pogramas BASIC es la memoria: tenemos una máquina con 128, 256, hasta incluso 1024 o en el colmo del derroche 4096 kas (¿naranja?); pero sólo podemos usar 23K, y si usamos Turbo-BASIC ya es el colmo de la claustrofobia: sólo tenemos 10K.

NestorBASIC alivia en parte este poblema, pues si bien no podemos hacer turbo bloques más gansos, sí podemos usar la memoria mapeada para guardar datos. Useáse que la consigna a partir de ahorina es:

NO GUARDES EN TU PROGRAMA BASIC LO QUE PUEDAS GUARDAR EN LA MEMORIA MAPEADA

Useáse que a partir de ahora está "prohibido" (tú eliges el grado de prohibición) meter a saco en tus programas:

- *Cadenas.* A nada que te descuides puedes perder 3 o 4K sólo con PRINT"Bienvenido a mi megafantabuloso programa que se hace cargo él solito", PRINT"Introduce el nombre del fichero a borejamear" y similares.

- *Datos de coordenadas, combinaciones de teclas, etc...* al estilo "if x>34 and x<100 then 1000 else if x<34 then..."

- *DATAs.* Es igual de fácil hacer un bucle de READs que hacer un bucle de lectura desde la memoria mapeada con NestorBASIC. Bueno, casi.

También podemos ahorrar memoria BASIC si hacemos algunas tareas en ensamblador, por ejemplo leer el ratón y los cursores y comprobar si el cursor/puntero se sale de los márgenes de la pantalla; lógicamente estas rutinas en ensamblador también estarán en la memoria mapeada y no en la idem BASIC.

Y tú pensarás "también ahorro memoria si no pongo comen-tarios". Y lo que más mola es que esto ya lo hace NestorPreTer... ju ju...

Una última considerancia desas: ya que vas a usar la memoria mapeada, antes de programar nada coge papel y hoja y apunta cuántos segmentos vas a necesitar, y el "mapa" de cada uno de ellos, es decir las direcciones en las que pondrás cada cosa. Esto facilita mucho la labor posterior y tal.

Asín pues vamos ya con trukis más conkretos a propósito de lo explicacionado hasta aquí.

NestorTRUKI el #1: COMPARTIENDO VARIABLES

Esto no es etsxactamente un truki pa ahorrar memoria, pero también mola & puede ayudarte en múltiples circunscha[illegible]lis.

Que sí, que vale, que NestorBASIC tiene funciones para almacenar programas BASIC en la memoria mapeada, que mola un mazacote; pero puede ser que por cualquier razón no puedas (te faltan segmentos) o no quieras (es mu complicao pa tí) usar esta featura, y nesecites cargar otro programa BASIC con el LOAD de toda la vida pero sin perder las variables. Ahorina te etxsplico cómo hacerlo.

El truki es tan simple como coger y guardar las variables en la memoria mapeada, cargar el nuevo pograma, y recuperarlas. Dicho así suena muy simple, y en realidad lo es, siempre que tengamos todas las variables metidas en una matriz; por eso mesmo definimos la matriz D en el **listado 1**.

Explicación más detallística: si metemos todas las variables en una matriz lo que ganamos es que todas ellas están almacenadas consecutivamente en memoria, de forma que podemos copiarlas fácilmente a cualquier segmento de RAM usando la función de NestorBASIC para copiar bloques de memoria entre segmentos (la 10). ¿Y cómo sabemos dónde guarda el BASIC la matriz? Pues pa eso tenemos la instrucción VARPTR. Añade a esto que el número de segmento asignado a la memoria BASIC es el 255, y yastatoa: **listado 2a**.

Una vez que tenemos las variables a buen recaudo, cargamos el nuevo programa con un LOAD corriente y zahiriente, volvemos a definir D, recuperamos las variables y a vivir que son 34 eones: **listado 2b**.

Inciso: tras vipalizar el listado 2b, a lo igual tú estás diciendo "¿Ans? Yo compré el NestorKit#1 en la XVI ru de BCN, y en las instrucçaos del NestorPreTer no ponía nada de ficheros externos de macros..." Aru de aru, ¡es que la versión que tú tienes ya está obsoleta! Tras la ru he desenvolupado una nueva versión, que admite la posibilidad de usar un fichero externo con macros; de esta forma no hace falta definir las macros que sean comunes a varios listados en cada uno de ellos (por ejemplo las funciones de NestorBASIC), basta con definirlas una vez en el fichero externo. No [illegible]ores, tú pide por esa boquita y te enviaremos la versión nueva...



NestorTRUKI el #2: SOY UNA PERSONA COORDENADA

Vamos a suponer este supositorio: estás haciendo tu editor de algos contento y feliz, y por sus bodrios es un editor en modo gráfico con sus iconos y demás croquetas. Digamos que tienes 34 iconos y una rutina de muevimiento del puntero; llerbonses, cuando detectas una pulsación del botón ratonil, has de comprobar sobre qué icono se ha pulsado, para actuar consecuentemente. ¿Cómo lo faces?

La maformanera más normal (en realidez no se me escurre otra) es tener una tabla con las coordenadas iniciales y finales de cada icono y dadas las coordenadas actuales del ratón, recorrer dicha tabla comprobando si las coordenadas roedoras se encuentran dentro de rango de coordenadas de cada icono (uséase si el icono va de (X0,Y0) a (X1,Y1) hay que comprobar si X0<X<X1 y Y0<Y<Y1). Lógico, ¿no?

Lo que distingue al ser rematadamente obsoleto del Auténtico Usuario de NestorBASIC (TM) es la maforma de hacer esto. Un pobre cualesquiera que no haya pasado del GOSUB lo hará a saco en BASIC, guardando en DATAs las coordenadas de cada icono. ¡Por favor, pero qué desperdicio de memoria! Suerte que pertenecemos al grupo de los AU-NB, y vamos a realizar dicha tarea de la forma más elegante y educada posible: en ensamblador.

Pues sí, en un segmento de RAM guardaremos una rutina en ensamblador junto con las coordenadas de todos los iconos habidos y por haber. Al llamar a dicha rutina con la función 59, simplemente le pasaremos las coordenadas actuales del ratón, ella buscará en la tabla y nos devolverá el número del icono que se ha pulsado. ¡Más economía memorística imposible! Y encima tenemos la ventaja añadida de la rapidez del ensamblador. Soy la ley.

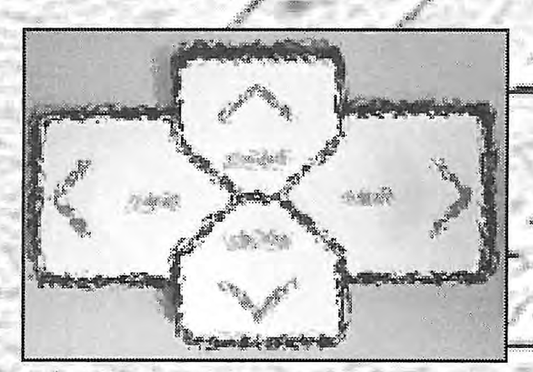
Asín pues, hete aquí tres listados más (more). El **listado 3** es una rutina universal para cargar un fichero en un segmento, es la que usaremos para cargar la rutina CM + las coordenadas (cuando digo que es universal no quiero decir que es made in Júpiter, sino que la usaremos también en todos los ejemplos que requieran cargar un fichero). El **listado 4a** es el idem en ensamblador del buscador de iconos (nombre no muy apropiado pero en fin, te aguantas), y el **listado 4b** es el cacho de BASIC que se encarga de llamar a la rutina cargada con el 3. Bon proveito.

NestorTRUKI el #3: NADAR Y GUARDAR LA ROPA

Este no es un truki de esos supermegahiperlarequeteidem, y de hecho es más bien una mariconada, pero te servirá para que tus programas se distingan por una inusitada elegancia y estilo propio de los más educados programadores y bla, bla...

Al guano, digo, al grano: Inate, por decir algo, que tu pograma se desarrolla en SCREEN 0, 80 columnas, y color blanco sobre fondo negro. Además, para tener más espacio pantallil haces un KEY OFF; y para más tranquilidad auditiva desactivas el sonido de las teclas. Falen, mola, pones todo eso al principio y yastá. Pero, ¿qué pasará cuando tu pograma termine? Pues que ese será el estado en el que quedará la pantalla; si resulta que el usuario es el Tàpies o el Mariscal y tenía SCREEN 3.14, COLOR 2,3, WIDTH 4, y la macarena cada vez que se pulsaba una tecla, pues se lo has desconfigurado todo al pobre ohnvre.

Solución: guardar el estado inicial de la pantalla, que averiguaremos consultando algunas variables de sistema, antes de establecer el modo pantallil que nos interese (**listado 5a**). Asín, al terminar podremos dejarlo todo como estaba (**listado 5b**). Y por supuestísimamente, lo meteremos todo en la memoria mapeada, que por algo somos los mejores y los más guapos (sobre todo yo, claro).



NestorTRUKI el #4: ME PULSE CUALQUIER ESPACIO

¡Uró! Este truki es muy parecido al #2. Si usamos el ensamblador para detectar fácilmente iconos, ¿por lo cualo qué no facer lo mesmo para detectar pulsaciones de teclas? La idea es simple: en vez de usar INKEY$, INPUT$ y obsolescencias similares, guardamos en memoria mapeada una tabla de teclas + una rutina en ensamblador que detecte qué ídem (o combinación de) está siendo pulsada, y nos devuelva su número asociado.

Ventajas: pues como en el caso de los iconos, ahorro de memoria BASIC (basta un simple USR para recorrer toda la tabla y obtener la idem/ combinación pulsada), rapidez, y facilidad para detectar teclas normalmente no muy detectables, como SHIFT, CTRL, SELECT, ESC...

Ahora bene, ¿cómo se guarda una "tabla de teclas"? Como ya sabrás (o no), desde código máquina el teclado se ve como una matriz de 11 filas por 8 columnas, de forma que cada tecla tiene asociado un número de fila y un ídem de columna. La susodicha matriz ya la puse en el SD#9 y creo que en algún número más, pero bué: **tabla al uso**.

La rutina ensambladera que usaremos (**listado 6a**), para simplificar las cosas sólo detecta pulsaciones de teclas sueltas, o bien en

combinación con SHIFT y/o CTRL; por tanto, para cada tecla habrá en la tabla de idems: número identificador, combinación con SHIFT/CTRL requerida, fila y columna. Por lo demás, el sistema operativo se encarga de escanear el teclado en cada interrupción del reloj (50 ídems por) y dejar información sobre el estado de cada tecla en una zona de la ídem de trabajo: usaremos esta información para determinar qué tecla está siendo pulsada.

Ah sí, se me esparcía: además de la tecla/combinación actualmente pulsada, la rutina también devuelve la ídem que estaba siendo idemada en la anterior llamada a la rutina. Esto es muy útil para detectar las pulsaciones una sola vez, aunque la tecla continúe pulsada: para ello daremos la detección por buena sólo si la tecla está pulsada Y no lo estaba en la anterior llamada. También puedes detectar cuándo se suelta la tecla (si no está pulsada pero sí lo estaba en la anterior llamada).

Y por sus idems, en el **listado 6b** tienes el correspondiente ejemplo en BASIC. Hala, ya podeis dejar de aplaudir.

NestorTRUKI el #5: LA BIOS TAMBIEN QUIERE JUGAR

Mucho hablar de usar rutinas ensambladeras propias, pero hemos olvidado que disponemos de un buen arsenal de idems ya hechas y listas para usar en la propia ROM del ente obsoleto: la BIOS. Y tú dirás: "Sí, pero la BIOS ya se puede usar con DEFUSR y no hace falta el NestorBASIC ni ná". Y yo, tras patearte las encías, te respondo: "Aru de aru de e, pero con DEFUSR no puedes establecer los registros de entrada ni consultar los de salida, cosa que sí puedes hacer con la función 58 de NestorBASIC."

Manque muchas de las rutinas de la BIOS no son sino versiones código máquina de instrucciones BASIC, hay algunas que nos pueden resultar bastante útiles. Ahí están sin ir mas lejor CHGCPU (&H0180) y GETCPU (&H1083), que respecticionalmente cambian el procesador del Turbo-R y obtienen el procesador actual. ¿Qué cómo se face ello? Pues como dice el **listado 7**.

Más ejemplos: la rutina CHGCAP (&H0132), que permite cambiar el estado del led del CAPS (**listado 8**). O la rutina GETPLT (&H0149), esta vez en la SUB-BIOS, que te permitirá obtener la paleta de un determinado color (**listado 9**). Como ves hay miles de premios: de ti depende rastrear el listado de la BIOS en busca de la rutina perdida, aquella que es la única capaz de alegrarte el día (japonesas aparte).

NestorFIN: ACABARAMOS DE AQUI

Espero que hayas aprendido la lección: sólo NestorBASIC puede hacerte realmente feliz (japonesas aparte, claro). Ademore el hecho de poder ampliarlo fácilmente con rutinas maquineras propias o ajenas espanta sobremanera el perenne fantasma de la obsolescencia. Que se note.

¿Comentarios? ¿Sugerencias? ¿Críticas? ¿Hemorroides? Ya sabes: estoy aquí para complacerte (en el buen sentido (de nuevo, japonesas aparte)), uséase que ya puedes ponerte a escribirme una carta/emilio manque sólo sea para decirme: "Estás zummmmbao". Hala, hasta. Os deseo que.

A borejam si se piensa el Puto que él es el onlyco que puede poner tekistos "ofderrekor" en sus artículos. Ná, que me predoneis por ponere los listados tan desordenoides y apelotonados,, peroske si no, el artículo ocupaba 34 páginas, que manque parece que mola, resulta que no. Por cierto, ahora que el Ramoni ya no es el capitán, nadie me recorta los artículos. Omola.



Konami Man

```
'///
'/// Listado 1: Carga de NestorBASIC
'/// y ejemplo de código NestorPreTer
'/// En resumen: que NestorPreTer
'/// transforma este listado
'/// en un idem BASIC de toda la vida.
'///

'**********************************
'*                                *
'* Primero definimos las macros *
'*                                *
'**********************************

'— Constantes —

@define TRUE -1
@define FALSE 0
@define SI @TRUE
@define NO @FALSE
@define ON @SI
@define OFF @NO

'@REQ_SEGS define el número mínimo de segmentos
'que necesitaremos, incluyendo los cinco primeros,
'que son usados por el sistema:
'Cuando NestorBASIC es cargado, se comprueba
'si hay al menos REQ_SEGS disponibles; en caso
'contrario se muestra un mensaje de error
'y NestorBASIC es desinstalado.

@define REQ_SEGS 6

'— Variables —

'Centralizaremos todas las variables en una matriz D,
'que será creada tras cargar NBASIC (ver truki el #2).
'Los contadores pa bucles han de ser variables simples.
'También definiremos la varable simple @ERROR para
'recoger el idem de las funciones de NestorBASIC.

@define NUM_VARS   2  'El número de variables
          'que utilizaremos (aparte de @ERROR y
          'los contadores de bucle)
@define ERROR        e
@define NUM_SEGS   d(0) 'Número de
          'segmentos disponibles
@define FILE_HANDLE d(1)  'Identificador
                           'para un fichero

'...
'define aquí todas las variables que vayas a usar
'y no olvides establecer NUM_VARS adecuadamente
'...

'Definimos también los contadores de bucle
'que usaremos

@define BUCLE b
@define BUCLE2 b2

'— Funciones de NestorBASIC —

'Usando macros para las funciones de
'NestorBASIC nos evitamos el engorro de tener
'que usar USRs y recordar numeros de funciones

@macro NB_UNINST e=usr(0)
          'Desinstala NestorBASIC
@macro NB_INFO     e=usr(1)
          'Obtiene info de NestorBASIC

'...
'define aquí todas las funciones que vayas a usar,
'o bien defínelas en un fichero externo de macros
'(lo cual es más cómodo)
'...

'— Ejemplos de macros útiles —

'Limpieza del búfer del teclado:

@macro CLEAR_KEY_BUFFER defusr1=&h156:
                         @ERROR=usr1(0)

'Comprueba si se está pulsando ENTER
'(if @PULSA_ENTER then...)

@macro PULSA_ENTER (peek(&HFBEC)
                     and 128) = 0

'Desinstala NestorBASIC liberando la memoria
'BASIC, y termina
'OJO: No uses esta macro desde un turbo bloque

@macro FINISH p(0)=@SI: @NB_UNINST: end

'— Para identificar el listado —

@remon
'list1.bas
@remoff
```


Pobre hombre...

```
'*********************************
'*                               *
'*   Carga de NestorBASIC        *
'*                               *
'*********************************

~       maxfiles=0:      'Esto ahorra unos
                         '250 bytes
        keyoff: screen 0: width 80:
        color: color 15,0,0:
        ?"— Cargando NestorBASIC... —": ?:
        bload"nbasic.bin",r:
        defint @ERROR:         'La variable de
                         'error de NestorBASIC
        @ERROR= 0:
        if p(0) >4 then
                goto ~LOADOK 'No hay error si
                             'P(0) es al menos 5
        else
                ?"ERROR: ";

'*** ¿Error? Pues mostramos mensaje y acabamos.

~       if p(0)=0 then
                ?"¡No hay memoria mapeada o
                             sólo hay 64K!":
                end

~       if p(0)=1 then
                ?"¡Error de disco al cargar
                             NestorBASIC!":
                end

~       if p(0)=2 then
                ?"¡No hay segmentos libres!":
                end

~       if p(0)=3 then
                ?"NestorBASIC ya estaba
                             instalado.":
                @NB_INFO:
                goto ~ALR_LOAD

~       if p(0)=4 then
                ?"Error desconocido.":
                end

'*** Salta aquí si NestorBASIC se ha cargado bien.

~LOADOK:        ?"NestorBASIC cargado OK!":
                ?"Segmentos disponibles:"; p(0): ?
```

```
'*** Salta aquí si NestorBASIC ya estaba cargado.

~ALR_LOAD

'*** Comprueba si tenemos al menos REQ_SEGS
'    segmentos, si no termina.

~       defint d: dim d(@NUM_VARS):'Crea
                                'matriz de datos
        if p(0)< @REQ_SEGS then
                ?"ERROR: ¡No hay suficientes
                         segmentos libres!":
                ?"Tengo"; p(0) ;" segmentos y
                         necesito al menos";
                @REQ_SEGS; "segmentos.":
                @FINISH
        else
                @NUM_SEGS= p(0)
'...
'Pon tu programa a partir de aquí.
'Sugerencia para empezar:

_turbo on (p(), d(), e) dim f$(1): defint @BUCLE:
                        @BUCLE=0: @BUCLE2=0
```



¿Tiene sueño, o es que es japonesa?

```
'///
'/// Listado 8: Cambio del led del CAPS
'/// con CHGCAP
'/// Funcionamiento: poner en A el valor
'/// 0 (led ON) o cualquier otro (led OFF)
'/// antes de llamar a CHGCAP
'///

@define OFF     1
@define ON      0

@define BIOS  e=usr(58)
@define CHGCAP       &H132

~       p(2)= @ON*256:          'o @OFF
        p(0)= 0:
        p(1)= @CHGCAP:
        @BIOS
```

```
'///
'/// Listado 2a: almacenamiento de las
'/// variables (la matriz D) en un
'/// segmento de memoria mapeada.
'/// Ejecútesemele antes de cargar otro
'/// programa BASIC. Suponemos que
'/// las variables son enteras.
'///

@macro LDIRSS e=usr(10)      'Transferencia
                             'de un segmento a otro

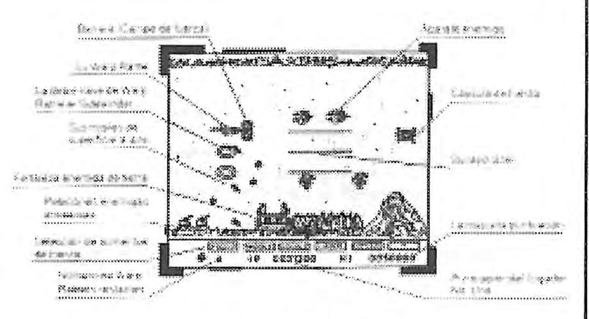
'Segmento y dirección donde guardaremos las
'variables: defínelas a tu gusto y conveniencia, yo
'pongo valores cualesquiera.

@define DATA_SEG 6
@define DATA_DIR &H100

'Guardamos las variables. Se supone que has
'definido NUM_VARS adecuadamente
'(en un listado como el idem 1, por ejemplo):

~       p(0)= 255:  'Segmento de la mem. BASIC
        p(1)= varptr(d(0)): 'Dir. en memoria BASIC
                            'de la matriz
        p(2)= @DATA_SEG:
        p(3)= @DATA_DIR:
        p(4)= @NUM_VARS * 2:  'Cada variable
                              'entera son dos bytes
        p(5)= @NO:   'O @SI, esto lo estableces
                     'a tu gusto
        p(6)= @NO:   'Lo idem de lo mismo
        @LDIRSS:

        run "elotro.bas"   'Ya podemos ejecutar
                           'otro programa BASIC
```



Manual del auténtico Konami Man

```
'///
'/// Listado 2b: Recuperación de las
'/// variables guardadas en la memoria
'/// mapeada con el código del listado 2b.
'/// Ejecútesemele al principio del
'/// programa que ha de recuperar las
'/// variables guaradadas por su predecesor.
'///

@remon
'elotro.bas
@remoff

@define LDIRSS e=usr(10)

'Segmento y dirección donde están guardadas las
'variables:de nuevo, defínelas a tu gusto, pero ojo,
'¡dales el mismo valor que tenían en el listado 2a!

@define DATA_SEG 6
@define DATA_DIR &H100

'Lo primero es volver a definir P() para que
'NestorBASIC chute.

~       defint p: dim p(15): define @ERROR:
                                @ERROR=0:

'Creamos D de nuevo, se supone que en el nuevo
'programa también has definido NUM_VARS.
'Consejo: define DATA_SEG, DATA_DIR y
'NUM_VARS en un fichero externo de macros, y
'usa este fichero con la opción /MAC al "compilar"
'todos los listados. Así te evitas tener que
'definirlas en cada uno.

        dim d(@NUM_VARS):

'Ahora que tenemos otra vez D disponible,
'recuperamos las variables:

        p(0)= @DATA_SEG:
        p(1)= @DATA_DIR:
        p(2)= 255:
        p(3)= varptr(d(0)):
        p(4)= @NUM_VARS * 2:
        p(5)= @NO:
        p(6)= @NO:
        @LDIRSS

'...y a partir de aquí, la vida sigue:

_turbo on (p(), d(), e) dim f$(1): defint @BUCLE:
                      @BUCLE=0: @BUCLE2=0
'etc...
```

```
'///
'/// Listado 3: Carga de un fichero
'/// completo en un segmento
'/// (el fichero ha de ser de 16K como
'/// máximo, ofcoursemente)
'/// NOTA: @ERROR y @FILE_HANDLE
'/// estan definidas en el listado 1
'///

@macro F_OPEN e=usr(31)   'Abre un fichero
@macro F_CLOSE e=usr(32)  'Cierra un fichero
@macro F_READ e=usr(33)   'Lee de un fichero

'Segmento y dirección en la que cargaremos el
'fichero, definir al gusto. FILE_SIZE es la cantidad
'de bytes que intentaremos leer del fichero.

@define FILE_SEG 6
@define FILE_DIR 0
@define FILE_SIZE 16384-@FILE_DIR

'Nombre y ruta del fichero a cargar

@define FILENAME "c:\rutinas\aluso.bin"

'Abrimos el fichero, si hay error saltamos a una
'rutina ~ERROR que se supone ya te definirás a tu
'gusto

~Listado_3:
        F$(0)= @FILENAME:
        @F_OPEN:
        if @ERROR<>0 then ~ERROR else
        @FILE_HANDLE= p(0)

'Ahora intentamos leer 16K del fichero. Si
'obtenemos el error 1 o el 199 (para DOS 1 y DOS 2
'respectivamente) lo ignoramos, ya que ese error
'significa simplemente "Final de fichero encontrado",
'es decir, el fichero es más pequeño que 16K

~       p(0)= @FILE_HANDLE:
        p(2)= @FILE_SEG:
        p(3)= @FILE_DIR:
        p(4)= @FILE_SIZE:
        p(6)= @NO:
        @F_READ:
        if (@ERROR<>0 and @ERROR<>1 and
                @ERROR<>199) then ~ERROR

'Ya está, ahora simplemente cerramos el fichero
~       p(0)= @FILE_HANDLE:
        @F_CLOSE:
        return
```

```
'///
'/// Listado 4b: Uso de la rutina
'/// buscadora de iconos desde
'/// NestorBASIC, de forma práctica,
'/// rápida, corta y elegante. (-v-)v
'///

@macro ENSAMB_EXE e=usr(59)

'Primero cargamos el fichero con la rutina en un
'segmento cualesquiera.
'OJO: La dirección de carga (@FILE_DIR) ha de ser
'la misma que la dirección en la que la rutina ha
'sido ensamblada (directiva ORG del listado 4a):

@define ORG_DIR &H8000

@define FILE_SEG 6
@define FILE_DIR @ORG_DIR
@define FILE_SIZE 16384-(@ORG_DIR-
                            &H8000)

~       gosub ~Listado_3

'...
'... Código al uso en el que, por ejemplo, detectamos
'... la pulsación del ratón en las coordenadas (X,Y)
'...

'Ahora averiguamos a qué icono corresponden
esas coordenadas:

~       p(3)= x:
        p(4)= y:
        gosub ~CALL_ENSAMB:
        if p(5)= -1 then ~NO_ICON
        else on p(5) gosub... 'O bien: goto
                              '~PROCESA_ICON

'Subrutina para llamar a la rutina en ensamblador

~CALL_ENSAMB:
        p(0)= @FILE_SEG:
        p(1)= @FILE_DIR:
        @ENSAMB_EXE:
        return
```

Esto es para que los de XSW se pregunten qué demonios pone aquí

```
;— Listado 4a: Buscador de iconos
;  en ensamblador.
;  Dadas unas coordenadas, recorre
;  una tabla de idems correspondiente
;  a las posiciones de sendos iconos, y
;  devuelve el numero del icono que
;  contiene las coordenadas entradas.
;  Entrada: P(3) = BC = Coordenada X
;           P(4) = DE = Coordenada Y
;  Salida:  P(5) = HL = Icono
;           encontrado (-1:ninguno)

        org     #8000

        push    bc,de   ;Para que no modifique
                        ;P(3) ni P(4)
        call    CHKICON
        pop     de,bc
        ret

CHKICON:
        ld      ix,TABLA_ICON

BUCLE:ld        a,(ix)
        cp      #FF
        jr      z,FINICON

        ld      l,(ix+1)
        ld      h,(ix+3)
        ld      a,c
        call    RANGE
        jr      nz,NOICON

        ld      l,(ix+2)
        ld      h,(ix+4)
        ld      a,e
        call    RANGE
        jr      nz,NOICON

SIICON:
        ld      l,(ix)
        ld      h,0
        ret

NOICON:
        inc     ix
        inc     ix
        inc     ix
        inc     ix
        inc     ix
        jr      BUCLE

FINICON:
        ld      hl,-1
        ret
```

```
;— Subrutina: RANGE
;   Comprueba que un byte esta dentro de un rango
;   ENTRADA:  H = Valor superior del rango (inclusive)
;             L = Valor inferior del rango (inclusive)
;             A = Byte
;   SALIDA:   Z = 1 Si esta dentro del rango (Cy = ?)
;             Cy= 1 si esta por encima del rango (Z = 0)
;             Cy= 0 si esta por debajo del rango (Z = 0)
;   MODIFICA: AF

RANGE:
        cp      l       ;Menor?
        ccf
        ret     nc
cp      h               ;Mayor?
        jr      z,R_H
        ccf
        ret     c

R_H:    push    bc      ;=H?
        ld      b,a
        xor     a
        ld      a,b
        pop     bc
        ret

;— Tabla de coordenadas de los iconos
;  Formato: identificador + x inicial + y inicial
;  + x final + y final (1 byte cada uno)

;  OJO: El identificador #FF esta reservado como
;  marca de final de la tabla (obligatoria)

;  Suponiendo que todos los iconos estan situados
;  en una matriz rectangular a partir de la posicion
;  base (BX, BY), y que tienen un tamanyoTX x TY,
;  se puede usar la macro "icon":

BX:     equ     0       ;Valores de ejemplo
BY:     equ     200
TX:     equ     11
TY:     equ     11

icon:   macro   @num,@xi,@yi
        db      @num,BX+@xi*TX,BY+@yi*TY,
        BX+@xi*TX+TX-1,BY+@yi*TY+TY-1
        endm

;Ejemplo: tabla de 5 x 2 iconos:
TABLA_ICON:
_0:     icon    0,0,0
_1:     icon    1,1,0
_2:     icon    2,2,0
_3:     icon    3,3,0
_4:     icon    4,4,0
_5:     icon    5,0,1
_6:     icon    6,1,1
_7:     icon    7,2,1
_8:     icon    8,3,1
_9:     icon    9,4,1
FIN:    db      #FF
```

```
'///
'/// Listado 5a: guardamiento del estado
'/// de la pantalla en un segmento
'/// Me apalearán por esto, pero acepto
'/// el sacrificio (qué remedio)
'///

@macro R_SEGI e=usr(3)    'Lectura de un
                          'segmento con autoincremento
@macro WSEGI e=usr(7)    'Escritura idem

'Segmento y dirección en la que guardaremos el estado
'Lo de siempre: que te las definas como te salga
'del teclado

@define STA_SEG 6
@define STA_DIR 0

@define VARIABLE v    'Para un bucle al uso

'Las variables de sistema al uso

@define LINLEN &HF3B0    'WIDTH actual
@define CRTCNT &HF3B1 'Líneas en pantalla
@define CLIKSW &HF3DB 'Sonido de teclas,
                       '(0=no, otro=sí)
@define CNSDFG &HF3DE 'KEY ON (0) / OFF (otro)
@define FORCLR &HF3E9    'Color de texto
@define BAKCLR &HF3EA    'Color de fondo
@define BDRCLR &HF3EB    'Color de borde
@define SCRMOD &HFCAF    'SCREEN actual

'Guardamos el estado con un bucle chapucero
'(ya sé que he dicho antes que es mejor no usar
'datas para ahorrar memoria, pero se supone que
'esto lo vas a poner en un pograma corto de
'inicialización, además no son muchas, y si no ya lo
'mejorarás tú, ¡y deja de quejarte ya!)

~       p(0)= STA_SEG:
        p(1)= STA_DIR:
        restore ~VARIABLES

~       read @VARIABLE:
        if @VARIABLE<> 0 then
        p(2)= peek (@VARIABLE):
        @W_SEGI:
        goto ~~

~YASTA: 'Aquí la vida sigue...

'Hete aquí las variables usadas, es importante
'respetar el orden en la línea DATA
~VARIABLES: data @SCRMOD, @LINLEN,
        @CRTCNT, @CLIKSW, @CNSDFG,
        @FORCLR, @BAKCLR, @BORCLR, 0
```

```
'///
'/// Listado 5b: Restauración del
'/// estado inicial guardado con el
'/// listado 5a
'/// OJO: ha de ser ejecutado fuera de
'/// turbobloques para evitar hemorroides
'/// con algunas instrucciones incompatibles
'///

~       p(0)= STA_SEG:
        p(1)= STA_DIR:

        @R_SEGI:        'SCREEN
        screen p(2):

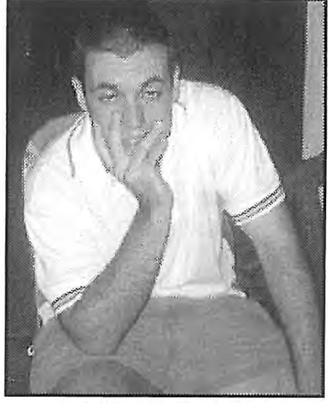
        @R_SEGI:        'WIDTH
        width p(2):

        @R_SEGI:        'Líneas en pantalla
        poke @CRTCNT, p(2):

        @R_SEGI:        'Sonido de teclas
        if p(2)=0 then screen ,,0
        else screen ,, 1

~       @R_SEGI:        'KEY ON/OFF
        if p(2)=0 then keyon
        else keyoff

~       @R_SEGI:        'Colores
        color p(2):
        @R_SEGI:
        color ,p(2):
        @R_SEGI:
        color ,,p(2)
```



NestorBASIC despierta pasiones

```
;— Listado 6a: Comprueba la pulsacion
;   de una tecla determinada
;   Entrada: -
;   Salida:  BC = P(3) = Combinacion
;                 pulsada (-1 = ninguna)
;            DE = P(4) = Combinacion
;                 pulsada en la
;                 anterior llamada

        org     #8000

;NEWKEY es la zona de trabajo en la que se
;guarda el estado de las teclas.
;Consta de 11 bytes, cada byte es una fila,
;y para cada byte, cada bit es el estado de la
;tecla para esa columna (1=no pulsada, 0=pulsada)

NEWKEY: equ #FBE5

CHKEY:
        ld      ix,TABKEY

BUCKEY:
        ld      a,(ix)
        cp      #FF
        jr      z,NOKEY

        ld      e,(ix+2)
        ld      d,0
        ld      hl,NEWKEY
        add     hl,de
        ld      a,(hl)   ;A = Estado de la fila
        cpl

        ld      b,(ix+3)
        inc     b
BUCFIL:
        srl     a   ;Cy = Estado tecla (1=Pulsada)
        djnz    BUCFIL
        jr      nc,NEXTKEY

        ;Comprueba SHIFT y CTRL:
        ld      a,(NEWKEY+6)
        cpl
        and     3
        cp      (ix+1)
        jr      nz,NEXTKEY

OKKEY:
        ld      c,(ix)   ;Pulsada?
        ld      b,0
        ld      de,(OLDKEY)
        ld      (OLDKEY),bc
        ret

NEXTKEY:
        inc     ix        ;Siguiente combinacion
        inc     ix
        inc     ix
        inc     ix
        jr      BUCKEY

NOKEY:
        ld      bc,-1     ;Ninguna combinacion
pulsada
        ld      de,(OLDKEY)
        ld      (OLDKEY),bc
        ret

OLDKEY:
        dw      -1  ;Para guardar comb. anterior

;— Tabla de teclas
;   Formato: identificador + SHICT + fila + columna
;   (cada uno un byte)
;   SHICT = &B000000CS
;   C=1 si ha de estar pulsada CTRL
;   S=1 si ha de estar pulsada SHIFT
;   Dicho de otro modo:
;   SHICT=0 para tecla suelta
;   SHICT=1 para tecla + SHIFT
;   SHICT=2 para tecla + CTRL
;   SHICT=3 para tecla + SHIFT + CTRL
;   Ver tabla 1 para la fila y columna de cada tecla

;OJO: el identificador #FF esta reservado como
;marca de final de la tabla (obligatoria)

TABKEY:         ;Tabla de ejemplo
        db      0,0,7,2 ;ESC
        db      1,2,7,2 ;CTRL+ESC
        db      2,3,7,2 ;CTRL+SHIFT+ESC
        db      3,1,7,2 ;SHIFT+ESC
        db      4,0,6,5 ;F1
        db      5,3,6,3 ;CTRL+SHIFT+CAPS
        db      #FF
```



"Te compras un sombrero y resulta que es una mierda..."

```
'///
'/// Listado 6b: Detecta el estado de las
'/// teclas en la tabla de ejemplo
'/// del listado 6a (detecta cuando se
'/// pulsan y cuando se sueltan)
'///

@macro ENSAMB_EXE  e=usr(59)

'Primero cargamos el fichero con la rutina.
'Ver nota sobre ORG_DIR en el listado 4a

@define ORG_DIR &H8000

@define FILE_SEG 6
@define FILE_DIR @ORG_DIR
@define FILE_SIZE 16384-(@ORG_DIR-
&H8000)

~       gosub ~Listado_3

'Croquetas inicializadoras

~       screen 0,,0:
        width 40:
        color 15,0,0:
        keyoff

'Creamos una tabla de cadenas para mostrar
'información sobre la tecla 'que se pulsa.
'Ya sé que he dicho que no mola guardar cadenas
'nel pograma BASIC, pero,
'¡¡JOER QUE SOLO SUN EJEMPLO!!

@define CADENAS c$

~       dim @CADENAS(6):
        @CADENAS(0)= "ESC":
        @CADENAS(1)= "CTRL+ESC":
        @CADENAS(2)= "CTRL+SHIFT+ESC":
        @CADENAS(3)= "SHIFT+ESC":
        @CADENAS(4)= "F1":
        @CADENAS(5)=
"CTRL+SHIFT+CAPS"

'Bucle infinito que detecta y muestra las teclas

@macro BEEPEA       beep:beep:beep

~INFINITO:
        gosub ~CALL_ENSAMB:

        if p(3)=-1 and p(4)<>-1 then 'Se suelta
                                        'una tecla?
        ?"Suelta ";@CADENAS( p(4) );" !!":
                @BEEPEA else

        if p(3)<>-1 and p(4)=-1 then 'Se pulsa
                                        'una tecla?
        ? @CADENAS( p(3) ): @BEEPEA

~       cls:
        goto ~INFINITO

'Subrutina para llamar a la rutina en ensamblador

~CALL_ENSAMB:
        p(0)= @FILE_SEG:
        p(1)= @FILE_DIR:
        @ENSAMB_EXE:
        return
```

```
'///
'/// Listado 9: Obtención de la paleta
'/// para un color con GETPLT
'///

@define COLOR c
@define RED r
@define GREEN g
@define BLUE b

@macro BIOS  e=usr(58)
@define GETPLT      &H149

'GETPLT funciona de la siguiente manera:
'Entrada: A = Color a consultar (p(2)=AF)
'Salida:  BC = p(4) = VERDE + 256*AZUL
'           +4096*ROJO

~       @COLOR= 7 'Por ejemplo
        p(0)= 1: 'Esta vez es de la SUB-BIOS
        p(1)= @GETPLT:
        p(2)= @COLOR*256:
        @BIOS:

        @RED= (p(3)\4096) and 7:
        @GREEN= p(3) and 7:
        @BLUE= (p(3)\256) and 7:
        print @COLOR; "=";
            @RED, @GREEN, @BLUE
```

```
'///
'/// Listado 7: Obtención y
'/// establecimiento del procesata del
'/// Turbo-R
'/// Greetings to: BIOS
'///

@macro BIOS e=usr(58)        'Función para
                    'ejecutar rutinas de la BIOS

@define GETCPU      &H180
@define SETCPU      &H183
@define Z80   0  'Los posibles modos de CPU
@define R800ROM 1
@define R800DRAM    2

'MSXVER=Versión del MSX: 0=MSX1, 1=MSX2,
'2=MSX2+, 3=MSX Turbo-R

@macro MSXVER       peek(&H2D)

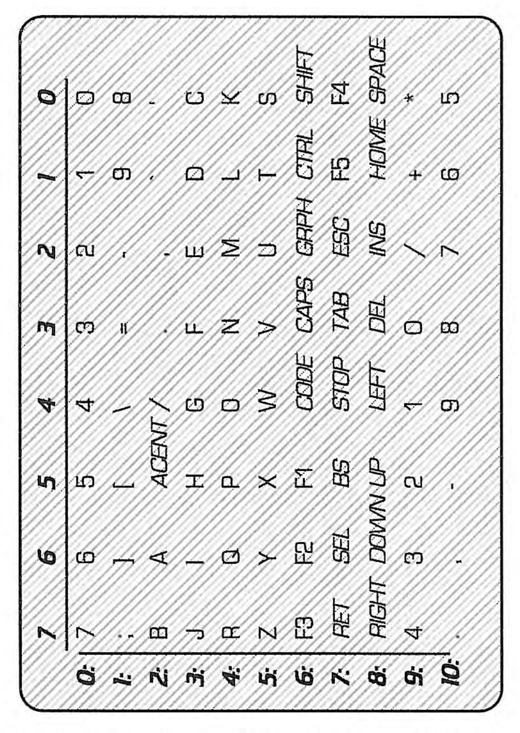
'Obtención de la CPU actual: tras llamar a
'GETCPU, en p(2) tendremos AF; el modo está en
'A, por tanto:

~GETCPU:
        if @MSXVER<3 then ~NO_TURBOR
else
        p(0)= 0:
        p(1)= @GETCPU:
        @BIOS:

        p(2)= (p(2)\256) and 255: 'Ojo, división
                            'entera (barra inv. o yen)
        if p(2)= @Z80 then...
        else if p(2)= @R800ROM then...
        else...

'Establecimiento de la CPU: para poner 0, 1 ó 2 en
A, pondremos &H8000, &H8100 ó &H8200 en
AF, o sea en p(2). El 8 es para que se actualize el
led adecuadamente; si lo cambiamos por un 0, no
se actualizará.

~SETCPU:
        if @MSXVER<3 then ~NO_TURBOR
                else
        p(2)= @R800ROM: 'o cualquier otro
        p(2)= p(2)*256:
        p(2)= p(2) or &H8000: 'Si queremos que
                            'se actualice el led
        p(0)= 0:
        p(1)= @SETCPU:
        @BIOS
```



	7	6	5	4	3	2	1	0
0:	7	6	5	4	3	2	1	0
1:	;	]	[	\	=	-	9	8
2:	B	A	ACENT	/	.	,	`	'
3:	J	I	H	G	F	E	D	C
4:	R	Q	P	O	N	M	L	K
5:	Z	Y	X	W	V	U	T	S
6:	F3	F2	F1	CODE	CAPS	GRPH	CTRL	SHIFT
7:	RET	SEL	BS	STOP	TAB	ESC	F5	F4
8:	RIGHT	DOWN	UP	LEFT	DEL	INS	HOME	SPACE
9:	4	3	2	1	0	/	+	*
10:	.	,	-	9	8	7	6	5

<u>Tabla al uso:</u> *fila y columna para cada tecla en la matriz del teclado. Las filas 9 y 10 se refieren al teclado numérico. Lo he puesto torcido por chulería y por falta de sitio.*

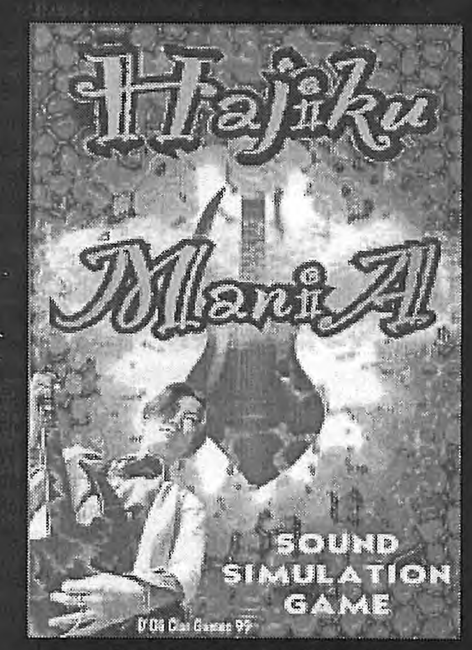
Hajiku ManiA

D'Oli Clar Games 1999

Hajukun Bodris comentarium tekisfile

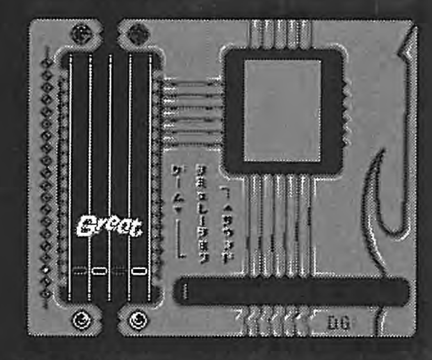
Hola payios y payias de todol mundo meskiskero. Es un placer como exagerao de grande presentaros este segundo juoco (acabado) de mi propia haciencia.

Muchos lo conocereis porque fue presentado en la pasada ru de Basuralona (Nestor antoni, me ponga que ru era). (La 16). Para los que no, aquí teneis este articulamen a modo de abórejam-ambórejam.



Portada del manualinbodris

Antes os cuento un poco cómo surgió la idea del juego ya que no fue mia, en resulta que el año pasado cuando Japolandié me encontré con que una de las novedades que había presentado Konami era una Guitarra Freaka. Me explico: El grupo de BEMANI (los autores de BEAT MANIA) habían sacado tres nuevos juegos casi al mismo tiempo, de los cuales uno se trata de un simulador de guitarra. Supongo que si no lo habeis visto nunca (lógico porque aún no está en España) os puede parecer tan raro como que el Antonio ya lo haya hecho 4 o 5 veces, pero os aseguro que es una pasada de lo wonder que mola el guason. La mecánica es muy parecida a BEAT MANIA, es decir, tenemos una partitura en pantalla que hay que tocarla tan bien como podamos mientras la música nos acompaña.



Pues llerbonses cuando yo vi eso me enladorié tanto tanto que en Japón empecé a hacer el programa de la versión MSX en papel, así hasta el dia que tuve una versión más o menos acabada, coincidiendo con la vuelta a España, entonces aproveché las 14 horas de avión para testear el programa haciendolo funcionar sobre las hojas. Os podeis imaginar lo duro que fue, no por mí -Yo estaba todo cachondo- sino por la Holís que estuvo a mi lado con cara susto todol viaje.

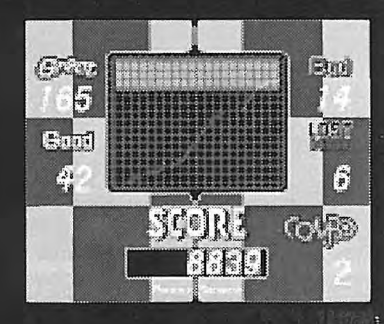
Lo que voy a decir a continuación hará que el Antonio me invite a lo menos a una coacola nel Ambodris: he de reconocer (lo siento por mi familia pque podra estar leyendo esto) que utilicé el Nestor Antoni Basic a na Mandi para desde Basic acceder al OPL4 y así poder tocar las notas por separanbodris. Lo hice, lo hice... no tenía otra opción.

Olvidemos las mariconadas varias y hablemos del juego pero ya.

Hajiku Mania es la versión MSX de Guittar Freaks, ya que al no haber ningún otro programa de este estilo pues lo considero asinnes. Hajiku Mania es onlicamente para MSX Turbo R con OPL4 y un Jardis Duro o similar. La base del juego es exactamente la de la máquina. A la izquierda de la pantalla aparece la partitura en tiempo



real de la música que tenemos que tocar, a la derecha tenemos un vidio Musicalizable y en la parte inferior una barra indicando el porcentaje de la música que estamos tocando. La dificultad está en que si no tocamos la nota exacta en el momento exacto estaremos tocando algo tipo *música-SaveR* (o sea, ruido). ¿Y cómo se toca la partitura? Pues ahí está la gracia: se toca pues con una guitarra freaka!! O lo que es lo mesmo, una cosa con forma de guitarra que tienes 4 botones (uno para cada canal de la partitura) y una pua (para hacerlos sonar). Así es como se juega a este juoco. Por supuesto para los que no se atrevan a hacerse una siempre podeis usar el teclado o el Joystick, pero de todas formas el juego viene con un manual que explica detalladamente los pasos a seguir para hacerse una guitarra freaka de estas. No olvideis que jugar con la guitarra mola un mazo de galo y queda muy quedón (y sino fijaros en la foto de MKII -Nestor, ponla en el articulamen-).



Hajiku Mania cuenta con un total de 18 temas en OPL4 listos para tocar. Entre los cuales figuran autores de la talla de SHAN, Quix, Manuel Pazos, Konamiman, etc. y otros no tan buenos como Yo o el Ramonijo y un bonus de Chiho Nakama. Todas incluidas en un disco (musical) llamado *1st Mix Unwieldy Chanels.*

I	SAVER	6720
II	MARCE	5909
III	CATAO	5486
IV	SAVER	5442
V	SAVER	5442
VI	MARCE	5321
VII	MARCE	5263
VIII	SOOOTO	5256
IX	CATAO	5114
X	AITOR	5094

Name SAVER
5442
Your Best Score Was
6720

Tanto Hajiku Mania como Guittar Freaks están programados de forma que puedan ampliarse el número de temas musicales sin límite alguno (bueno sí, el de tu disco duro). Pero la versión MSX tiene además la ventaja de contar con un conversor de MBWAVE al formato de Hajiku Mania, de forma que puedes convertir tus propias músicas y tocarlas desde aquí. Incluso una vez convertidas pueden copiarse en disco y ejecutarlas en el Hajiku Mania del amigo. En estos momentos estoy preparando lo que será el 2nd Mix para Hajiku Mania.



También existe un editor de videos musicales donde podeis crear nuevos o modificar los existentes

para incluirlos en vuestros propios temas. El último cambio es que Konami decidió diseñar una guitarra con tres botones mientras que Hajiku Mania usa una de cuatro, ya que sino resultaba muy muy facil.

La puntuación que se obtiene en la máquina de Konami va directamente a su página WEB donde todos pueden ver a los mejores jugadores. Hajiku Mania no se ha sobrado tanto pero se lo monta de forma que, todas las puntuaciones son grabadas en HD subdivididas en mejor puntuación en cada tema, top 10 respecto a cada jugador y top 10 respecto a todos los jugadores.

Como nota final os comento que los otros dos juegos que diseñó Konami el verano pasado fueron DANCE DANCE REVOLUTION y DRUMS MANIA. El primero como su nombre indica es un "simulador de baile", cuenta con una mini pista sobre la que has de bailar colocando los pies tal y como muestra la pantalla. En noche vieja hice un prototipo de DANCE DANCE REVOLUTION: el Odoru Mania, pero con el inconveniente de que tenias que usar zapatos metálicos (un fracaso). El segundo juego es un simulador de batería. Sin duda alguna este es el mejor de todos, ya que cuenta con una batería incluída y suena realmente bien. Será mi próxima meta cuando acabe lo que tengo ahora entre manos: "*Tataku Mania*", que no es otra cosa la versión MSX de Beat Mania.

Entre tanta versión os informo que Beat Mania, Dance Dance Revolution y Guittar Freaks están disponibles en Play Station con sus respectivos aparatejos para hacerlo todo más real. Drums mania en cambio no lo está pero si aparecerá si no lo ha hecho ya para la Play Station 2se.

Pero toda esta movida está en Japón, y en España, ¿cuándo? Conformaros con ver las máquinas en algunos recreativos, pero en cuanto a los juguetitos...



SEX BOMB BUNNY

¡¡Omoló el dia que menteré quesa gente rara haría un juoco de **guarros**!!

Creo que me lo dijol L'**Antonin**. Yo no me lo creí porque yo soy **normal** y mis amigos que también lo son (sobre todo el **Ramboni**) tampoco se lo creyeron. Yo por mis partes se lo dije a mi mare garosha y me dijo que no me ajuntara con esos *extranyos*.

La primera estaje ya empieza fuerte

Y asin lo hice, porque yo he visto en un documental cosas de gente cabeza rapada que dan mucho miedo y pegan a la gente sin **pedir** permiso primero. Así que seguí jugando a juocos educativos y olvidé aquella noticia de muy mal gusto antes de que pudiera afectarme.

Pero un dia pasó por mis manos una afoto de los **fruitis** y me dije: Si estos que son tan raros resulta que son buena gente y a más es bueno pa los ninios, entonses los raros esos también puede que sean... **¿fruitis?**

Entonses cuando quise preguntarlo vi que a una casi se le salia un algo y pensé: "Esto no lo he visto antes en los fruitis", pero también como me había saltado algunos capitulos porque quería ver Ferdy es valiente (**¡Ferdy!**) pues igual por eso mo me acordaba. Na, que les dije hola y no me pegaron, entonces me di cuenta que eran pacificos y no querían matar a nadie, incluso la chica de las, ¡me dio un beso!

Entonces le dije a mi mala madre que si tienen **Mesekis**, les gusta el **Mazinjer Z** y se pintan la cabeza como los fruitis, la única persona rara era ella. No veais el susto que me llievé cuando comprendí tal cosa.

Llerbonses el juoco guarro lo han acabado, y como yo ya no les tengo miedo pues dije ¡adeborejam que lo quiero comentare yo! A más es un **jueco** que cuando lo ves con esos graficunis la musicasa **PSG** y tal parece del ochenta y algo, y te sientes como si de alguna maformanera hubieras



Foto mal hecha cogida de la home page

rescatado un clásico del Mesekis que no conocia nadie. Porque mola y poreso lo comento.

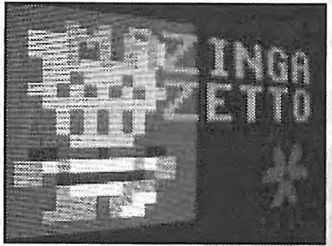
Para clasificarlo se puede decir que es un PACMAN pero de buen rollo (**PACGIRL**). Por cierto ahora macuerdo que cuanto estaba en Japón me hicieron una entrevista sobre cuánto conocía del PACMAN porque querían hacer una segunda parte a petición de los chinos. **Yo** le dije a la china

que era mu dificil hacerlo atrastivo si se limitiban a hacerlo en 3D (esa es la intención). Pero vamos a hablar del nuestro que mola más.

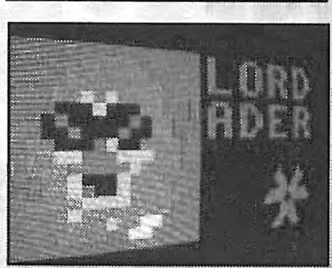


En lugar de coger bolas que eso está muy visto lo que hay cacer es bordear areas para así hacer desaparecer estas y aborejamear a la senyorita Seta Sero. **Ella** se piensa que ahí detrás está segura, lo cual le da confianza para estar a sus anchas y ponerse comoda. Lo que no sabe es que nosotros somos muy malos (...¿como se llamaban los malos de los fruitis?)

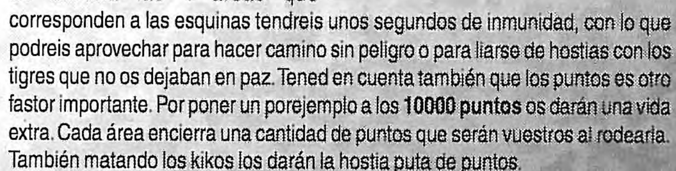
Nosotros somos el conejito de la suerte que va por ahí inocentemente sin saber que se esconde tras cada una de esas ventanitas.



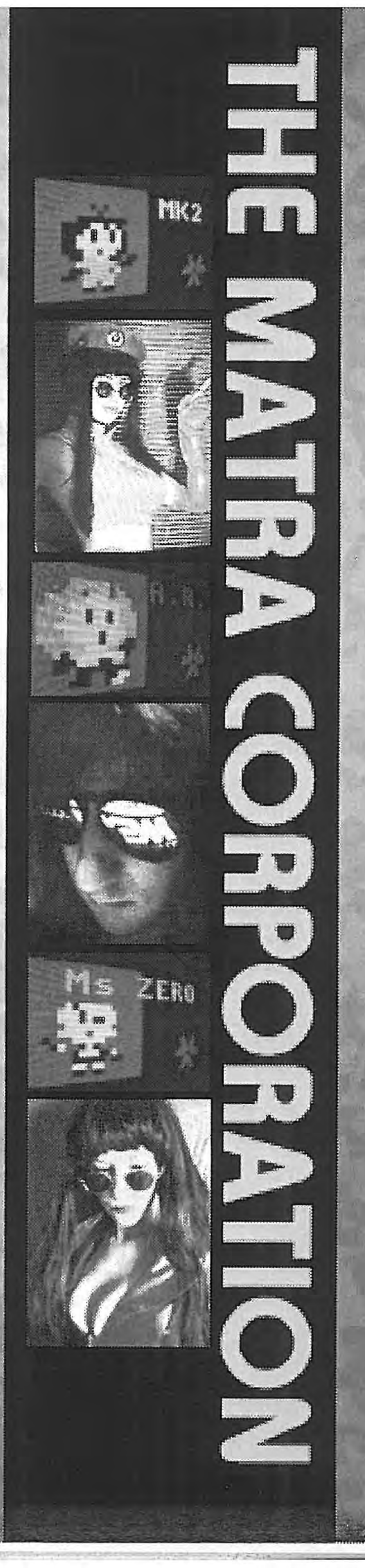
Sin embargo - esta es la primera vez que uso la palabra sin embargo - Sin embargo (la 2a) hay otros kikos como tigres o gatos que si saben lo que pasa. Y además están pagados porque si te fijas en la demo los malos bailan de cachondos que están porque la **Zezero** les ha dao un kisu. El gato es el que más jode porque corre más que nadie. Poco a poco el personal malo se incrementará y cada vez correrán más en una proporcion dificultad = nivel de pornografía de la estaje. Pero ya sabeis que tiran más dos desas que dos carretas. Asín que creo que no necesitais a nadie que os diga que lo intenteis una y otra vez más.



De todas maformas hay un truki que combiene que sepais: si os quitais de encima las 4 áreas que corresponden a las esquinas tendreis unos segundos de inmunidad, con lo que podreis aprovechar para hacer camino sin peligro o para liarse de hostias con los tigres que no os dejaban en paz. Tened en cuenta también que los puntos es otro fastor importante. Por poner un porejemplo a los **10000 puntos** os darán una vida extra. Cada área encierra una cantidad de puntos que serán vuestros al rodearla. También matando los kikos los darán la hostia puta de puntos.



He estado jugando bastante mucho pero aún no sé si es mejor ir a por todas y no hacer más que ir parriba y pabajo rodeando casillas o dedicarse a matar





Omola latikinbodris

gente. **Depende** bastante de cada fase, al principio en las dos o tres primeras si que va bien matar gente porque son fáciles y dan muchos puntos, pero luego sólo hay que buscar salidas y si por el camino destapas un algo pos es un detalle.

Lo que no podeis hacer es intentar destapar las zonas que más os "convienen" porque así siempre acabareis acorralados (lo digo por **Xperiencia**).

Y ya está, sería una tontería ponerse aquí a contar más el funcionamiento del juego, y ademas no sería propio de mí, asin que lo dejo.

Me reflexo:

Sin duda **SEX BOMB BUNNY** es una joya de juoco, y no me refiero a las imágenes de la **Zeni 0**, que están muy bien, sino a que es un juego como los de antes, un juoco "**de verdad**" al que no le falta nada. No tiene ningun efecto raro, usa el PSG... pero todo lo que tiene está realmente bien hecho. A veces no es necesario meter música OPL4 o scrolls sin sentido (¡un scroll sin sentido no es un texto que no se mueve!, sino efectos de relleno para tapar la verdad, que el juego es una mierda). Esta gente ha sabido hacer hasta un manual atractivo, que aunque no explica cómo jugar es una pasada en cuanto a colorido y maquetación. En mi opinión el juego les ha quedado redondo, y me gustaría que aquellos usuarios que programan echaran un vistazo a su acabado porque se puede aprender mucho de él. Además hay que tener en cuenta que es el primero que hacen y eso tiene mucho mérito. Esta gente sabe hacer juegos, y si no cambiar donde pone Matra por algo tipo **TAITO** o **SEGA**, ¿*alguien* lo notaría? Intentad ponérselo ahora al Pumpkin Adventure III (y a muchos otros). Estas empresas nos tenían acostumbrados a ver juegos tecnicamente mucho peores pero aún así nadie podría creerlo, ¿por qué? Simplemente porque no son atractivos. El uso de samples, música **OPL4** o imágenes en screen 12 no es ninguna ventaja si no se sabe hacer un juego.

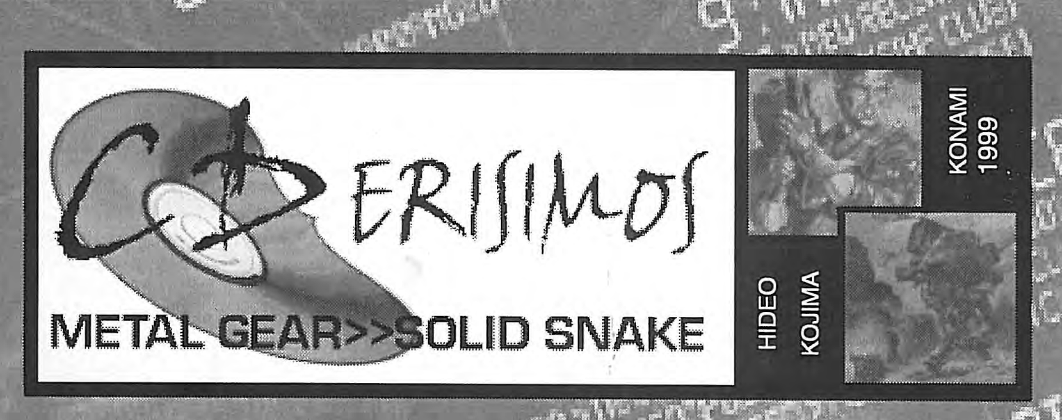
Nota: Si no tienes el juego te lo vas a comprar porque de lo contrario te podría resquebrajar en dos mitades y media y eso **duele** más que esquiar con la punta sobre una cuchilla dafeita*rrr*.



TEKISBODRIS

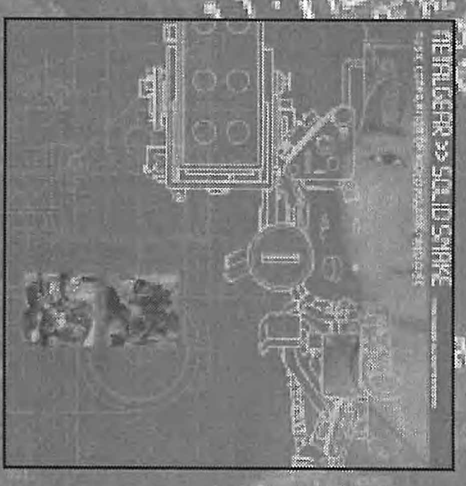
ABOREJAM QUE LAS AFOTOS TODAS TODAS LAS HEMOS COGIDO DE LA PAJA GÜEB DE LOS MATRAKAS (MATRA.CJB.NET). KE NOS PERDONEN LAS INCONGRUENCIAS PERO SOMOS COMO TONTOS. TAMBIEN HAY ESCANERS DEL KETEKET Y DEL MANUAL. SI NO LO HICIERAN TAN BIEN NO LO HABRIAMOS COPIADO. ¡AYOS!





Aquí vuelvo yo para comentar otra vez alguno de los CDrisimos más more wonders. Esta vez os voy a hablar un puquet sobre un nuevo CD de Metal Gear que me compré en mi pasado viaje a Japochina.

Como todos sabeis Hideo Kojima fue el payofón que dijo "Amos a facer un juoco de espionajeo guerrileo" y Konami le debió ver con cara chino y pensó que estaria loco. Pues este senyor que debe saber mucho (porque es muy feo) creó la versión meclese de Metal Gear en 1987 y luego para la NES un año más tarde. Aluego fue él mismo quien propuso hacer en 1990 Metal Gear 2 Solid Snake, y Konami dijo "Anda ke no". Se sacó ese mismo año otra versión para la NES tal que Snake's Revenge, y finalmente cuando ya era el más betos de todos dijo, y ahora vamos a hacer el Metal Gear Solid, cuando Konami dijo " ...Tardas".



Alesores todo esto lo dicho pa que os hagais cargo de esto: si Konami hizo tanta versión, ¿¿por qué se han decidido a hacer en 1999 un CD que se llama METALGEAR >> SOLID SNAKE ??, ¿por qué no aparece la version de Plallestation?, ¿y la NES?. Como veis Hideo Kojima (autor del CDrisimo) sólo ha incluido las versiones MSX, y así de claro está escrito en la carátula del CD. ¿Significa esto que sólo considera parte de la saga los dos juegos que se hicieron para MSX? Ahí queda eso.

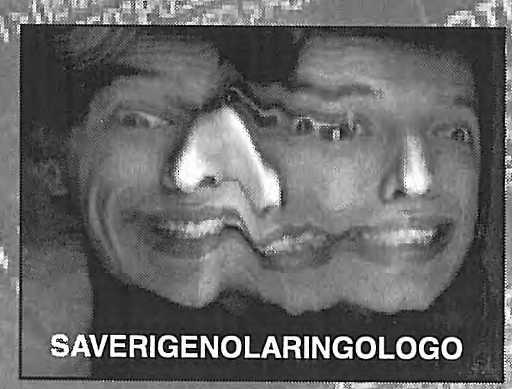
Entonces como acabo de decir este se trata de una recopilación de temas en plan transición del METALGEAR al SOLID SNAKE. Las músicas son las que todos conocemos pero con instrumentos "reales" aunque hay mucho sintetizador pero queda muy bien. De vez en cuando aparecen melodias que utilizan instrumentos clavados a los del SCC, y aunque os pueda parecer pastosen hacen un contraste que queda muy wonder. Eso sí, se trata de una versión bastante, no diría tecno porque no es tanto pero sí que usan muchos samples y baterias a toa hoxtia. Es una especie a aquella música que hizo Omega sobre Metal Gear en plan discotequera para OPL4.

De todas formas no son todas así, las lentas por ejemplo hacen mucho uso de los coros y otras tantas solos de guitarra, etc. No necesariamente os ha de gustar la música discotequera para que os llege este CD.

A continuación teneis una lista de todos los temas que aparecen:

1: TEME OF SOLID SNAKE
2: THEME OF TARA
3: THE FLONT LINE
4: FREQUENCY 140.85
5: SNEAKING MISSION
6: LEVEL 3 WARNING
7: RETURN TO DUST
8: CHASING THE GREEN BERET
9: IMMINENT
10: AN ADVANCE
11: ADVANCE IMMEDIATECY
12: NIGHT FALL
13: LEVEL 1 WARNING
14: -!- RED ALERT
15: INFILTRATION
16: FAREWELL
17: RETURN OF FOX HOUNDER
18: RED SUN
19: EXIT
20: HEAVY METAL GEAR

Antes de acabar os diré que el grupo de Hideo Kojima fue también autor de SNATCHER, SD SNATCHER y POLICENAUTS. Sobre estos tres existe otro CD recopilatorio de su invención. En este caso no parece tan evidente pero POLICENAUTS que en principio no guarda relación alguna con la saga SNATCHER sí forma parte del CD. Posiblemente pueda comentar este en el próximo númeramen, ya que depende de si voy o no voy este año a Japolandia. Soshite, ¡hasta!


SAVERIGENOLARINGOLOGO



En esta instantanea vemos el momento en el que el recién destituido Capitán Moncho es conducido hacia el exilio. ¿Qué opina el pueblo llano (y el esdrújulo) sobre la nueva situación política en el seno mismo del clups?


PROTOCHINO AL USO
"¡Tum tum pá!"


YOMBO CUALESQUIERA
"Ayokesé. Yopasso."


(¿EN?) TERAO SAN
"¿Pero no hablabais del Hideo Kojima? ¿No? Menos mal, porque yo soy otro chino."


EL INTERFECTO
"Nunca podré superarlo..."

MSX DEN-YŪ LAND

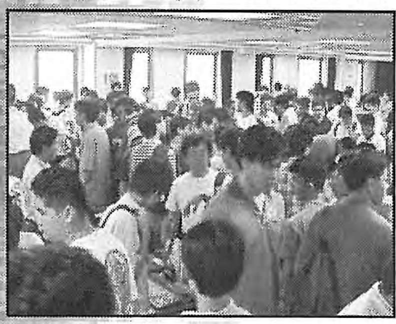
8/21-22 AKIBA

"No es por no hacer un MSX nuevo... si mira, ¡si es que ya lo tengo diseñado!, pero hacerlo, pa ná ...es tonteria." *-ASCII Corporation-*

Suncomentario igual comentario momentaneo mas un uno.

Es un complazco de granos el poder informaros de aquesta reunioncity tan aborejameantemente grandinbodris que se celebrón en Japón (China) el pasado 21 y 22 de augosto del naitinain.

Esta informacinbodris es un tekistexto que hemos recibido de nuestro amito Ikeda san. La ruñón tiene por nombre MSX DEN-YU LAND y es la firstera vegade que se hace, pero en este caso no afectó la ley primera vez=No vendrá



nadie, ya que en este caso acudieron al encuento más de 2200 Chinos I/O Japonbodris así como peces agordos de compañías como ASCII, Impuress, Hirose musen, LAOX, Compile y muchas otras. Estuvo también nuestro creador Nishi Ikeda de ASCII, Kitane (-LA- cabeza editora (HEAD EDITOR) de la MSX FAN), y Niitani (Presidente de Compile), más gente importante como Hirosawa de TAKERU, miembros de Panasonic, hasta el Sr. Manolo Pazoses acudió al encuentro... ...¿Qué pasó? ¿Es que se han vuelto locos los chinos? ¿¿Si no cómo sesplica que acuda tanta gentenbodris?? La respuesta está en que esta vez los organizadores han contado con un presupuesto de, atención: ¡¡un millón de Yenes!! ¿Cómo es pausible tal derroche? Pues facil, el dinero fue cedido por ASCII Education company, ASCII Future laboratory y Hirose musen itou pan. Nota: A mí no pregunteis quién es este último porque yo sólo estoy haciendo mi trabajo.



El caso es que fue mucha mucha gente y que también hubo mucho mucho material (normal, había que lucirse ante tanta gente). En total hubo 40 y pico stads de los que tenemos una lista-Pryca de lo que hubo y no hubo. Preparaos porque la pienso escribir a saco.

Lista-Pryca "de lo que" cedido por Kunichi Ikeda:

STUDIO Sequence

R2 CHASER (Shoot'em up): 300 Yen

MECHANICAL BRAIN (Idem) : 300 Yen

BATTLE MISSION (Idem de idem): 200 Yen

ASTRO MONSTERS (Idem x 3): 1000 Yen

Do-RAGON SOFT WORK

Algorythm (Un cderisimo de Audio)

TAKO-SYSTEM

Promotion Video tape

Hearthful disk

POPCoRN

SYSTEMAX

VAN! VAN! Sumou-kai



S+2

Shinya Sone mostró su MIDI System y software en su stand.

OSAKA Group

Keiichi Kuroda vendió muchas cosas de Meclese y sus Hnostars. Gracias!

Inaga an

HNOSTAR Japon

Ikeda y Fijiwara vendieron mucho soft y Hard Europeo en su stand. Pero sólo durante el dia 21 ya que para el 22 se les acabó el arsenal. ¡¡Menuda venta hicimos!!

HO-SOFT

Inuzorisha

SYNTAX

Su NV Magazine



Frontline

+PCM

+PCM Technical handbook

ESE RAM DISK Kit

Algunas revistas tipo Frontline magazine

Sensor kid: 1000 Yen

Mostraron también la promo de "PLEASURE HEART" para +PCM. ¡Se trata de un Shoot'em up fantastico! Pero encaretodavía no está acabado.

Uechan-dayo

"Welkis the ledgend" (Kakutou game like Over drive ...Me lo traduzcas Anton): 1000 Yen

Creo que es un buen juego, además Manuel Pazos compró uno. (Nota del traductor: ¿Y que pasa? ¿Es que porque lo compre el Sr. Manolo ya va a ser bueno? Me nieeego a comprarlo.)

GIGAMIX

Magical laberynth entre otros.

CTRL+G

Vendieron algunas MSX Magazines como son MSX-SEED y PoPiPu!, quizá las conozcas ya que en la revista Hnostar y XSW-Magazine se han escrito artículos sobre ellas.

IMORI PRODUCTION

Imopuro 3(Disk magazine): 300 Yen

ASCAT

MSX Technical guide book

MSX SOFTWARE COLLECTION (colección de materia Japonés de dominio publico)

Shirin Keibitai Yuuki (Juego de aventuras visual)

Dreams

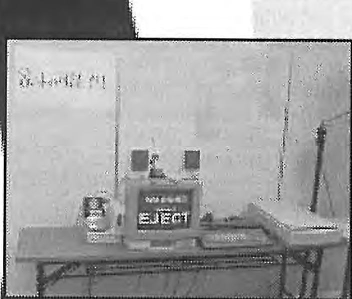
Prime factors (Otro Cderisimo de musicasas): 500 Yen

Selfish (Audio CD + Data CD): 800 Yen

Mo Soft *(Nota del traductor: ¡¡¡Las vacas locas!!!)*

Muchas magazines, entre ellas MAX.

Kyokugen: un MUY BUENO juoco de naves.



Koikeya

Satoshi Abe mostró su 3D CAR ACTION F-Nano2' en su stand. Con la nueva versión puedes llegar a conectar hasta 4 MSX Turbo R ¡¡via puerto del Joystick!!

ESE Artist's Factory

MEGA-SCSI

FAST RS-232C Cartidge HARUKAZE

ESE DSP Technical information (book)

Okada shuzou

Ellos vendieron mucho material MSX como MSX TR y F1 Spirit etc...

GENUINE NETWORK

Algunos CD de audio

La revista "Amateur NOW!"

Acrobat 232 : 2000 Yen

SS Pad adaptor: ¡Ahora puedes usar los pads de SegaSaturn en tu MSX!

Studio ESUTORIA

Algunos juegos como Pikachiu

LANDSCAPE

Niwatorisu promo: 200 Yen (Juego tipo tetris, ¡muy bueno!)

ALVA

NE'SHEPH (Una Mega demo made in Japan): 200 Yen

TPM.CO SOFT WORKS

PSG Fukyuu iinka (PSG Spread commission)

THUNDER FORTH (Cederisimo de audio): 700 Yen

Falcom no are (CD de músicas arregladas de Falcom)

MIDI DATA COLLECTION 1, 2, 3, 4, 5, 6 y 7: Cada uno a 300 Yen

Analogy

SONYC, NUTS, Hnostar, SD MESXES (¡nuestra visita española!)

Maple Yard

Moonlight Saga light version

Moonlight Saga Music CD

Moonlight Saga original T-Shirt

NIFTYSERVE MSX FORUM



¿Papá sólo tiene jerseises rojos?



Redactora Jefa de MSX FAN



Director de Compile

RCM

Yang vendió su Disk Magazine "EAST SEA" hecha en el Sur de Korea.

Especialmente ESE Artist' Factory y SYNTAX vendiren MUCHOS de sus productos, sé que llegaron a recaudar ¡más de 300000 Yenes! Ahora un Euro son 118.63 Yenes, entonces... ¡¡calculad 300000 Yenes!!

Para más información sobre la reunión dirigiros a:

MSX DEN-YU LAND Official web site

http://www.sh.rim.or.jp/~syntax/ev/index.htm

MSX Fairlist

http://www.msx4ever.demon.nl/fair.htm

Podrás también encotrar fotos en:

ZDNET GAMESPOT:

http://www.zdnet.co.jp/gamespot/news/9908/21/news01.html

AKIBA PC Hotline!:

http://www.watch.impress.co.jp/akiba/hotline/990821/etc_msx.html

PC Watch:

http://www.watch.impress.co.jp/pc/docs/article/990823/msx.htm

ASCII Corp:

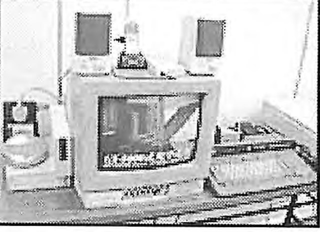
http://www.ascii.co.jp/ascii24/call.cgi?file=issue/1999/0825/hard01.html

Disfrutamos del meeting realizado por Nishi, Niitani (presidente de COMPILE) y kitane (La cabeza editora esa de MSX FAN ...Nestor (Néstor dice: EDITOR JEFE))

Nishi dijo "Ahora, yo estoy aquí. Este mensaje es para vosotros. Pero ahora yo no puedo hacer nada más..." Sé que Nishi quería contarnos el





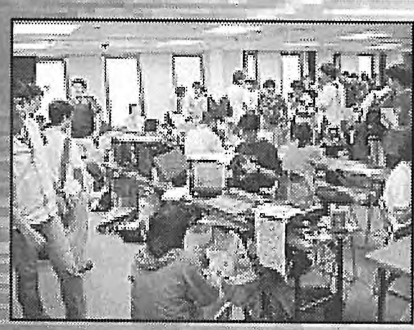


futuro del MSX, pero Nishi no puede contárnoslo, ya que en ese caso se vería atacado por ASCII, quienes no están muy de acuerdo en hacer un nuevo MSX, lo que supondría tener que desarrollarlo ellos mismos, y ASCII corporation no se encuentra en una buena situación estos momentos. Finalmente cabe la posibilidad de que los usuarios y otra gente no lo comprara.



Sabemos que a Fujioka (otro cabeza editor) de Weekly ASCII magazine no le gusta el MSX, pero nosotros queremos hacer una nueva máquina. ¡Quizá el nuevo MSX pueda usar la tecnología FPGA!

En cuanto a Niitami,



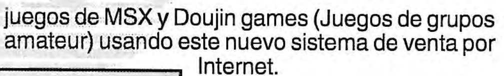


comentó que Compile quiere hacer ¡un NUEVO Disk Station ejecutable en los emuladores de MSX!

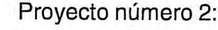
Aquí en Japón, Imamura de SYNTAX tiene dos grandes proyectos:

Proyecto número 1:

Un sistema de venta de Software al estilo de TAKERU pero vía Internet. Ahora Imamura está contactando con muchas SoftHouse. Quizá tengamos la oportunidad de comprar los viejos



juegos de MSX y Doujin games (Juegos de grupos amateur) usando este nuevo sistema de venta por Internet.



Proyecto número 2:

Imamura quiere hacer una revista de MSX. Ahora está contactando con ASCII Corp ya que quizá estos nos ayuden. Y yo debo escribir un articulo para esta nueva revista. Imamura quiere poner tambien la historia del MSX en Europa, así que, ¿quién tiene un artículo sobre la historia del MSX en ingles? ¡Si tú lo tienes, por favor ayúdanos!

Y aquí acabose todo de todo. Creo que es una buena forma de entrar en el año 2000, ¿no creeis? ¿Alguno de vosotros hubises imaginado tanta movida por estas fechas tras el abandono de la última empresa que desarrolló MSX en el 94?

Tekisbodris original: **Kunishi Ikeda**

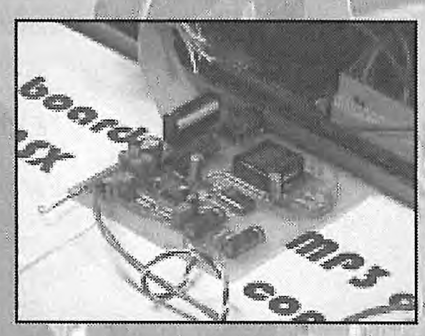
Traducción del tekisbodris: **SaveR**

Tilburguoya 2K

Otra más que ha conseguido pasar el efecto 2000







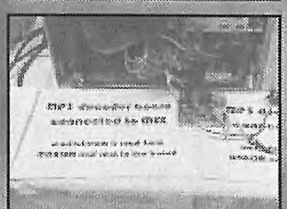
Anda que no mola el buen royo que hay con el mensaje de aquesto. Apareció por algún casual en la lista de correo y el Antonio va y dice de meterlo en el Fancinenbodris como si nos lo hubieran escrito para nosaltres. Pués, ahí queda eso. Os dejamos con el artículo de Ivan Isidro.



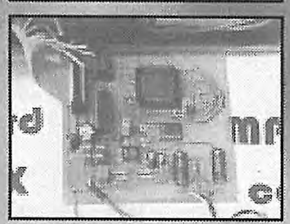


- Despegamos (con el avion, claro) con HORA Y MEDIA de retraso y bla, bla... (pasaremos el primer dia, que no viene a cuento...)

- Al dia siguiente, en la Reunion/Feria/RU de usuarios/amiguetes/otros, la cosa estuvo bastante animada. En una CASA de tropocientos metros metros cuadradados, repartidos por las habitaciones de sus 3 pisos, nos podiamos encontrar DE TODO: MSX de segunda mano: 8235 a tutiplen, 8245, 8250, algun MSX-1, impresoras, monitores, joysticks, placas, disketeras, grabadoras, etc, etc, todos ellos a bastante buen precio.







- En la parte de novedades, lo mas destacable es la presencia de los Japos: Kuniji Ikeda (que tio mas canijo y nervioso), acompañado por Ben, Nash, Tako, (y otro del que no me acuerdo del nombre), Ghost (presentando el juego Morning Star, mezcla de Arkanoid y Galaxian) y su novia ("My honey", como dice el, cosa que no me extraña). El grupo de Kuniji traia, entre otras cosas, ratones MSX de velocidad doble, algun Turbo-R, un MSX 2+ y otras cosas que no vendian, si no que subastaban!

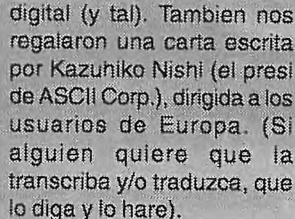


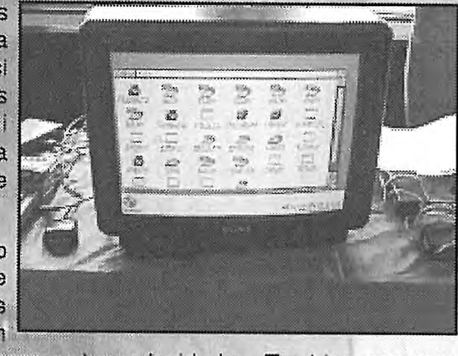
- Tambien presentaron el juego PLEASURE HEARTS. Un juego de scroll horizontal para MSX-2 (o superiores), que incorpora un nuevo sintetizador de sonido llamado "+PCM" (lease PRAS-PI-SI-EM, en Anglo-Japo), consistente en un DAC (COVOX) avanzado, para reproducir sonido









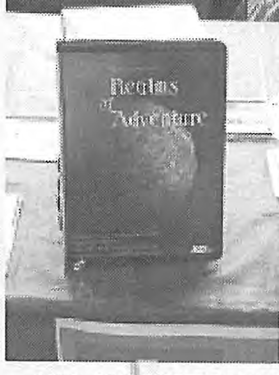


digital (y tal). Tambien nos regalaron una carta escrita por Kazuhiko Nishi (el presi de ASCII Corp.), dirigida a los usuarios de Europa. (Si alguien quiere que la transcriba y/o traduzca, que lo diga y lo hare).

- Padial, presentando (que no explicando, ya que me toco a mi hacer los honores) la placa Z380, con una nueva demo para comparar las velocidades. Tambien presento (primicia mundial) su nuevo expansor de 8 slots y DMA ¡FUNCIONANDO¡, tambien presento su cacharreria clasica. Curiosamente no vendio ninguna Z380, pero si algo de material menos novedoso.

- MSX Power Replay/Ramon Ribas, presentamos los ya conocidos CD recopilatorios XTORY, mas algun material de Italia y a las veces patrocinamos los productos de Padial. Ramon enchufo su Turbo-R y pusimos en marcha el HAJIKU MANIA, que por desgracia no era jugable ya que "alguien" no le mando la guitarra-joystick-interfaz para tocar. (Eso si, les pudimos dar bien la lata a los de Sunrise, con el "Smoke on the water" version Moonsound). Ademas Ramon se hizo con la distribucion del juego de Ghost, que viene presentado en un "envase" bastante original.









- Sunrise que, se colocaron enfrente de nosotros (o fue al reves??), tenian un stand bastate nutrido, tanto de personal como de productos (que nos encargamos de boicotear colocando un peazo banner de la Z380 encima de sus cabezas). Aun asi vendieron el nuevo IDE version 4.(algo), con o sin RS-232 y los RS-232 solitos (un poco mas baratos que a su precio normal), ademas de un nuevo juego-aventura al estilo del XAK (otro nooooo, por favor!). Tambien tenian demos del nuevo software titulador para G9000, entre otras cosillas. (Aproveche para encargarles las nuevas GAL para arreglar el antiguo IDE y ahorrarme unas pelillas).

- Los chicos de Futuredisk vendieron una especie de demo dedicada a la saga Metal Gear, ademas de vender el juego Thunderbirds.

- Los editores de la MCCM, su nueva revista electronica MCCW. Tambien pudimos contar con la presencia (y el honor) del fundador de dicha revista. Curiosidad: En

este stand se encontraba el/la usuario/a mas joven de toda la reunion, una chica que debia tener unos 11 o 12 años! (Para que luego digan que el MSX esta muerto...)



- En el atico, podiamos encontrar a Raymond y Mari (que es un chico, a pesar del nombre), pertenecientes al grupo MSX NBNO, editores de la revista XSW (por cierto, me regalaron el NUMERO 1 DE LA PRIMERA EDICION !!!). Tenian diversos juegos y programas para MSX,



entre ellos una especie de Puyo-puyo al que podias jugar. Tambien tenian manuales tecnicos del VDP-9958, del PSG, del Y-2413 (FM-PAC) y del Music Module (este ultimo me lo pille, por supuesto).

- Ademas de los stands de MSX, se pudieron ver algunos de no-MSX, que vendian de todo: Cintas para todo tipo de impresoras (excepto la VW-0020, snif !!), CDs guarretes, placas de PC viejas para desguazar, algun Atari, placas de MSX, carcasas de Philips, disketeras, material para impresoras de inyeccion de tinta, etc.

- Ya por la tarde (a eso de las 4 o 5), fuimos recogiendo y nos dirigimos al hotel. Por la noche, cenamos: Los Españoles (y Catalanes), los Japos y los Holandescos de Sunrise en el resturante Griego de costumbre (Rafa deja las pastillas de menta!) y algunos aprovecharon para hacer negocios (Hola Ramon).

- Al dia siguiente nos dimos una vuelta por Amberes en plan turistico, etc, etc. No puedo dejar de mencionar a Richard, un usuario de Luxenburgo!!! (debe ser el unico), que nos llevo de un lado para otro en su coche durante todo el viaje. Gracias a el pudimos visitar bastantes lugare y ganar algo de tiempo, ademas de pillar un gran dolor de cabeza gracias a la bomba de la gasolina de su coche.



Creo que no me he dejado nada por mencionar. La feria estuvo bastante animada (segun Rafa no estuvo tan bien como otros años), si bien el recinto era un poco inadecuado, ya que la gente estuvo repartida por varios pisos y habitaciones y era bastante jaleoso localizar a alguien en ese laberinto.

P.D.: Nestor, ya puedes dejar de babear !



XVI RU Baruseroneska

estoy delante dun mesexe...

Amos a ver: Queda una semana pa la próxima Ru, la XVII, y el Antonio, que ahora ques capitán se ha vuelto muy feo y muy agresivo, ma cogío por banda y amenazandome hasta dar miedo me ha sentado ante un 8245 con una cinta de la Ru pasada, la XVI, y el mando del vidrio pa que la comente...

Así por de pronto macuerdo de que fue el 4 de Diciembre [mentira, me lo ha dicho el Antón] y de que el sitio era el de siempre, las Cutcheres de Sans (o algo). También me acuerdo [o no] de que hubo bastante peña, unos 110 como va siendo habitual y, que, a pesar de los pesares, lo más mejor de todo fue el ambiente, la secta y las paridas del Mato con la videocámara.

LOS ESTANS:

Los estans eran muy majos, todos ellos, o a lo menos eso parece nel video...

MdeA: El están del señor Darriba es ya un clásico en las Rus, manque esta vez tambien era un poco clásico porque tenía lo mesmo de siempre: Compass, Animecha, Ruby & Jade, Amateurism, Notos, ... eso si, la novedad no pudo faltar: en esta ocasión fue el Sand Stone, tetris al uso made in Compjoetania TNG.


Stand del Mesexe

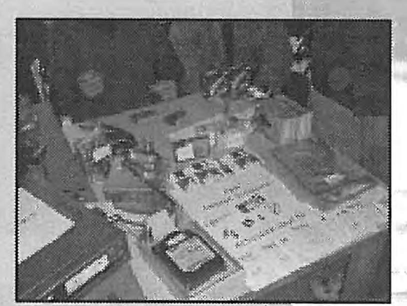

Zona fanzinera & Bra


Ninios enviciados. De mayores serán delincuentes

Matra El están de Matra era de los más repletos. Aparte del Navarone, invento de producción propia, y diversas cosiñas de producción nipona, la novedad era el Sex Bomb Bunny, que podía encontrarse en tres formatos: en Másquina recreativa, apañada para poder jugar todo aquel que lo quisiera, en formato Game Boy, ejecutado en una Super Nintendo y, como no, en versión MSX, que al fin y al cabo era la que se presentaba de veras. Además de todo esto se dejaron ver promos (o algo) del Moscow 2048 y varias muchas otras cosas que no salen en el vidrio.

MESXES Nosotros. ¿Qué hacíamos nostros allí? El SaveR presentó el Hayiku Mania, alias juego de la guitarra, un chorro de juegos japoneses y el Neoatoejejome 1.5. Sin duda lo más llamativo fue la guitarra, ya que se montó un concurso con premio incluido (una camiseta) que ganó Verónica Velasco. Más cosas que podían encontrarse en nuestro están: Las Nestorutilidades de turno, representadas esta vez por el NestorKit#1, el fanzain de turno, esta vez el SD#13. Por último los más atrevidos participaron en un Karaoke de Mazinger Z organizados por ya podeis imaginaros quien...

Lehenak Los amigos del Norte llevaban esta vez un nuevo número de su fanzain que, por cierto, a pesar de ya no ser el capitán me lo siguen regalando (omola!). Pero no solo de fanzain vive el mesxesiano. También necesita algo de violencia. Por eso tambien traian pistolas de Sega convertidas a MSX. Como curiosidad el MSX 8234 expuesto en este están. El MSX portátil más sectario de la Ru! Ah, y para los más mangueros otro mini manga de Verónica sobre ADN y el genoma humano... (que curioso, me parece que hace poco hablaban de esto en el telediario...)



Payofón arreglador de IDE's



Todos quisieron saludar al Ídolo.



34 MSX por metro cuadrado

Moai Mas fanzains! Y este de tamaño grande y con portada en color (plastificada!). Pero no solo eso, tambien el Nash, un juego erótico festivo conversacional de creación propia. Tambien junto a este están, un poco más a la derecha la división de hardware hacía las modificaciones al interfaces IDE de Sunrise en vivo y en directo.



Sora ni sobieru kuro ga neno shiro



Sus amigos lo abandonaron.

Están de todo lo vendo y todo lo compro Pos esta vez estuvo mejor. Dado el fracaso de la anterior Ru uno ya se pensaba que no quedaba nada por vender, pero no, esta vez la cosa se animó y volvió a verse actividad en el están. Esperemos que siga así porque mola ir comprando y vendiendo cosas como si fuera el mercadillo de



El MSX hombre está obsoleto



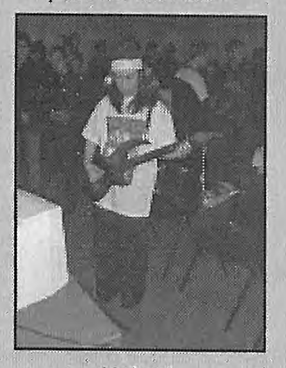
Presentando el nuevo juego Matrero.



¡ Puto !



¡ El / la / los / las !



¡ Holas !



¡ Celista !

los domingos.

Y había más, y más, los señores de CUOAM, Power Replay, el están de la AAM, y alguno que me dejo en el tintero.

LO DEMAS Como era de esperar la Ru terminó con el chino de la muerte de rigor donde nos hicimos cargo por enesima vez periscopeando las triboyas como ya es tradición.

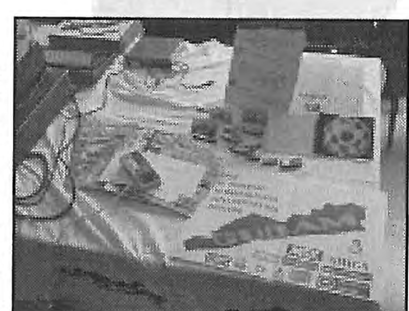
Bueno, y como ya no macuerdo de más, y me quedado sin vidrio donde mirar, pos os haceis una idea con lo que os he contado, y lo que no os lo inventais, que nadie se dará cuenta. Al fin y al cabo dentro de una semana nos veremos de nuevo en la Ru, la XVII...

Mola.

Ramón [postadata: Ale Antón, ya te he hecho el artículo. Ya puedes parar de darme latigazos... ...si te parece bien, vamos...]



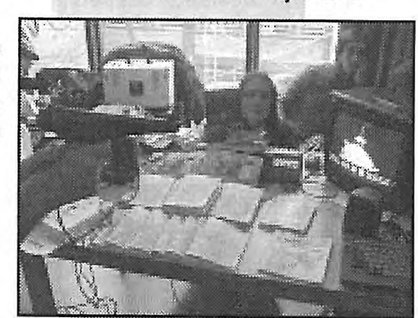




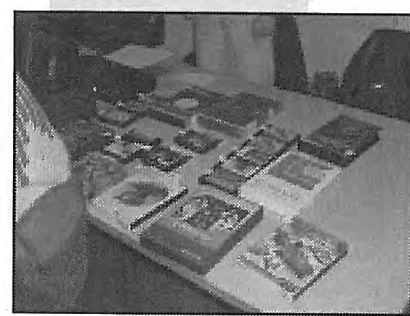
Stand e Guodam



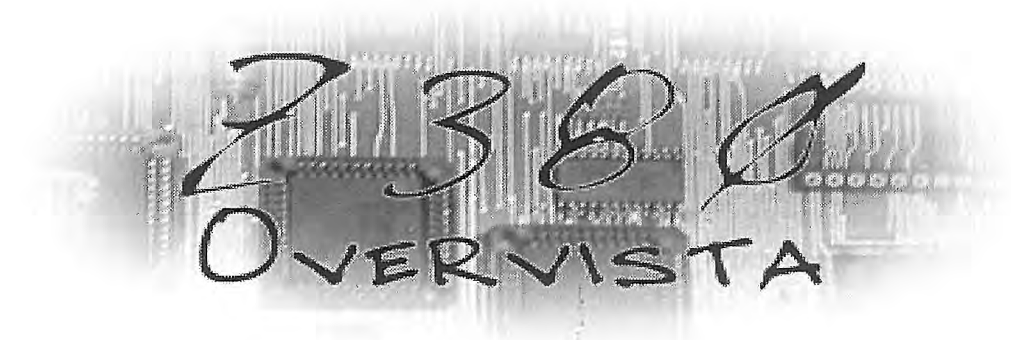
Piñando en el stand Lejeniano



Veronica Naranjo en su stand



El stand de 2dos guarros



PREFACIO

Hallábame yo en la pasada ru de obsoletos de Madriszxk, cuando acercando mi continente corporal hacia el stand de Leonardo Padial halleme de bruces ante un invento a primera vista curioso, y también significativo tras un análisis más detallado: la placa LPE-Z380. Inicié pues una conversación con el susodicho hombre, tal que:

- Hola buen hombre. Tu invento me interesa. ¿Cuántas unidades monetarias exiges a cambio de una unidad del mismo?

- Pues tal que 35.000 unidades de peseta.

- ¡Buenooo! Hállome en una difícil tesitura, pues mi poder adquisitivo no da abasto para formalizar la transacción. Sin encambio, puedo ofrecerte una entrada de 15.000 pesetas y el resto en cómodos plazos (cómodos para mí, claro). A cambio, me dedicaré en cuerpo y alma a investigar con fruición desmedida las posibilidades del engendro, a fin de desarrollar provechosos softwares de aplicación para el mismo.

- Acepto barco, pero deja de hablar así por favor...

FACIO

Muchos de vosotros habreis oído hablar de la tarjeta Z380 desarrollada por Leonardo Padial (nombre bonito: LPE-Z380), pero permaneceis en la más absoluta inopia en cuanto a los detalles sobre la misma. Pues tranquilos, he (medio) adquirido una y tras hacerle unas cuantas pruebecillas de nada, aquí teneis un simulacro de artículo que arrojará luz sobre vuestras dudas (espero). No daré detalles concretos de programación, pero tranquilos, está en preparación un completo manual en el que se explican todos los pormenores de la placa sin acudir al campotraviesa.

¿QUE ES?

Empecemos por lo más básico, useáse: ¿qué es exactamente la LPE-Z380? Pues ni más ni menos que una placa con un procesador Z380 a 14.38 MHz, que dispone además de su propia memoria. La LPE-Z380 está diseñada para tres modos de funcionamiento:

- En el *modo cartucho*, que hoy por hoy el más práctico, simplemente se conecta la placa a cualquier slot del MSX, actuando por tanto como un periférico más. En este modo, el Z380 de la placa y el Z80 del MSX ejecutan sus respectivos programas de forma

independiente y en paralelo, por lo que disponemos de una multitarea rudimentaria (podríamos llamarla "bitarea"). - En el modo reemplazo, también se conecta la placa a un slot del MSX, pero en esta ocasión el Z380 sustituye al Z80 del MSX. En este modo el Z380 trabaja a 3.57 MHz, ya que de lo contrario no podría acceder al hardware del MSX (que no soporta velocidades mayores); sin embargo la velocidad global del sistema aumenta, ya que el Z380 ejecuta las instrucciones de código máquina en menos ciclos que el Z80. La desventaja de este modo es que exige una modificación interna en el MSX: es necesario cortar un par de líneas del Z80, lo cual puede resultar bastante complicado con según qué modelos de MSX. - Si se conecta la placa al expansor de 32 bits de Padial, el Z380 se convierte en la CPU del MSX modular.

¡QUIERO VERLO!

Pues venga: en la **figura 34a** tienes un afoto de la placa y una somera descripción de sus componentes más importantes. La placa usa un módulo SIMM como memoria principal del Z380, por lo que es relativamente fácil y barato encontrar el módulo más adecuado a nuestras necesidades/bolsillos (la placa soporta SIMMs de capacidades entre 1 MByte y 128 MBytes). Ojo, ha de ser memoria EDO.

La memoria biport es la única forma de comunicación entre el MSX y el Z380: esta memoria puede ser accedida simultáneamente por ambos procesadores (el MSX la ve en el slot donde esté conectada la placa, el Z380 la ve en una zona determinada de su espacio de direccionamiento). Para la comunicación se usa un mecanismo de handshaking, que explicaré más adelante con un ejemplo. El tamaño de esta memoria es 1K; sí, es muy pequeña, pero es que este tipo de memoria es muy cara... hay que joderse...

El resto de componentes ya son croquetas: los PLDs son un par de chicharros de tipo FLASH (programables eléctricamente, aunque no desde el MSX) que controlan la lógica combinacional entre el Z380 y el resto del sistema; los jumpers permiten configurar el modo de funcionamiento y el tamaño de la memoria; y creo que no me dejo nada importante.

¿POR QUE?

Ya es la 1:41am, y antes de quedarme totalmente dormido voy a enumeraros algunas de las razones que hacen interesante la programación para el procesador Z380:

- El Z380 es 100% compatible con el Z80, por lo que podemos empezar a programar inmediatamente sin que sea necesario que volvamos a aprender ensamblador desde cero, ni variar sustancialmente nuestras técnicas de programación.

- Por otro lado, las nuevas instrucciones del Z380 nos permiten realizar programas de forma más fácil y optimizada. Ejemplos que hablan por sí solos: ADW HL,DE; EX HL,BC; EX B,B'; LD BC,IX; PUSH #1234; LD HL,(SP+#1234)... por no hablar de virguerías como las instrucciones de multiplicación y división.

- Los registros están extendidos hasta 32 bits, por lo que ya podemos hacer algo tan bestia como LD HL,#12345678 y quedarnos tan anchos. Además los registros están cuadruplicados: disponemos de cuatro juegos de registros principales + cuatro juegos de alternativos.



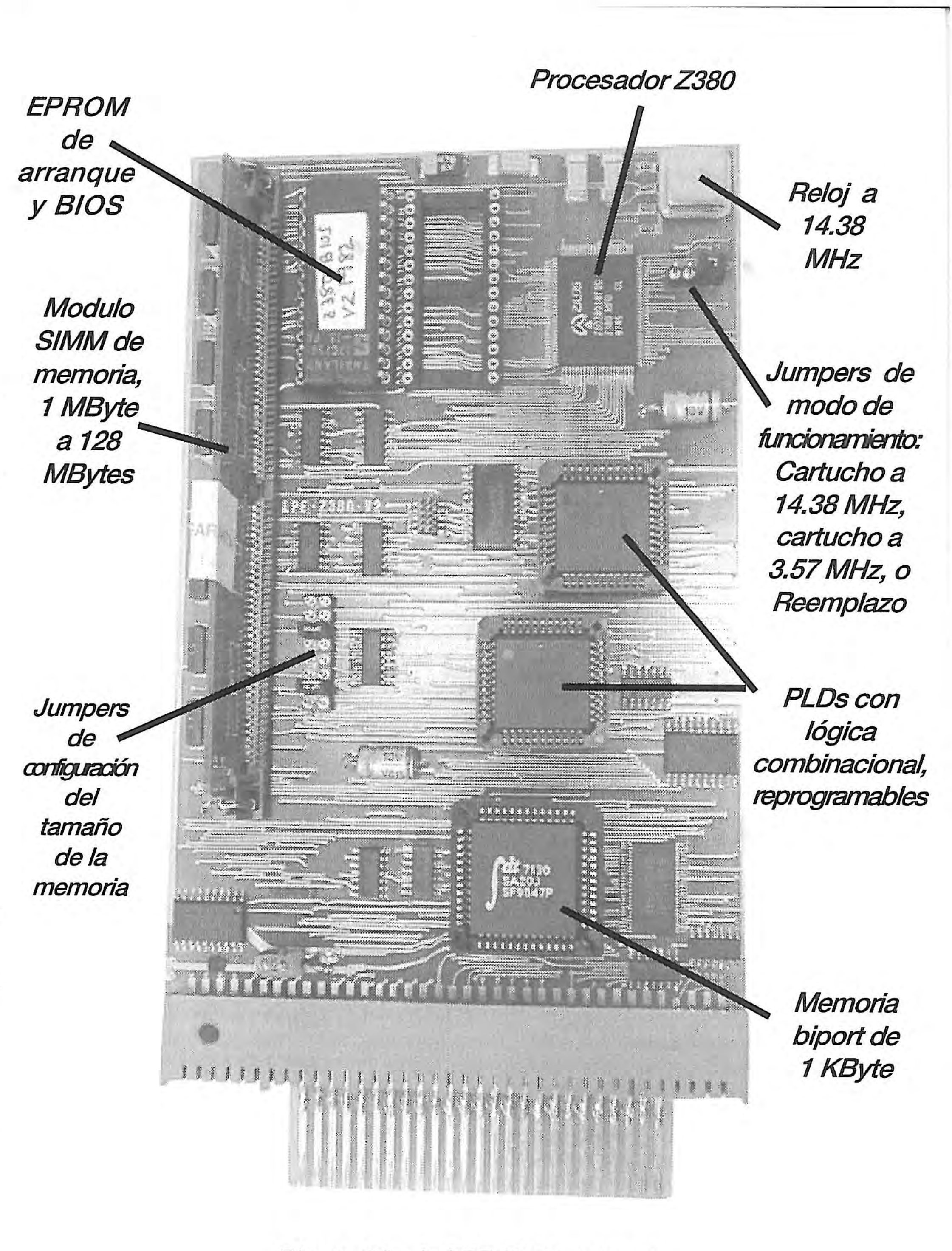


Figura 34a: la LPE-Z380, tal cual

- La memoria es totalmente lineal. ¿Os imaginais hasta 128 megas lineales, uséase, LD A,(#12345678)? ¡¡Anda que no mola!! Se acabaron las hemorroides con la paginación, los slots y demás mandangas.

Sin embargo no todo es perfecto en este mísero mundo. Hemos de tener en cuenta también que la placa presenta una limitación bastante importante: el Z380 sólo puede acceder a su propia memoria, y a la memoria biport que comparte con el MSX. Esto significa que no puede acceder directamente al hardware del MSX, y "hardware" significa: memoria, puertos, VDP, VRAM, disco, PSG, FM, joystick, ratón, rodón, teclado... vamos, todo lo que añadido a un Z80 forma un MSX.

El truki de la muerte para convertir un algo de 8 bits en uno de 32 (o al revés)

¿COMO?

Entonces, ¿cómo puede el Z380 comunicarse con el mundo exterior (el hard del MSX)? Pues mediante el handshaking explicado antes: ha de pedirle amablemente al MSX que acceda al recurso deseado, éste ejecuta la petición y le devuelve el resultado al Z380... todo a través de la biport, por supuesto. Y claro, previamente el MSX ha de estar muestreando constantemente si hay alguna petición pendiente por parte del Z380. ¿Complicado? No tanto: mírate los listados 34b y 34c, que se ejecutan simultáneamente en el MSX y en el Z380, respectivamente. Que conste que es sólo un "ejemplo académico", y por tanto 100% chapucero y sin tener en cuenta ciertos detalles de implementación tales que bla, bla...

Con esto debería quedar claro un punto importante: debido a la imposibilidad de acceder al hardware, a no ser que usemos el modo reemplazo (bastante más engorroso que el modo cartucho) el Z380 NO puede ejecutar directamente software del MSX, sólo funciona con programas específicos diseñados para la placa.

EL ESTADO DE LAS COSAS

De momento la cosa está un poco verde con respecto a la LPE-Z380. Me explico: aunque la cuestión del diseño y contrucción del hardware ya está finiquitada, no existe aún software para la placa aparte de unas cuantas rutinas de prueba, y lo peor es que tampoco existe ningún entorno de desarrollo para Z380. Por fortuna, la compatibilidad con el Z80 nos permite usar los ensambladores de siempre a costa de usar sólo instrucciones Z80 (también se puede rizar el rizo y emular las nuevas instrucciones del Z380 a base de macros, como hago yo, pero esto incrementa considerablemente el tiempo de ensamblado). Pero no tenemos ninguna forma de desensamblar/depurar los programas creados.

Pero en fin, si siempre nos pusiéramos a llorar no habríamos llegado a donde estamos ahora. Lo que hay que hacer es ponerse a hacer cosas, como podamos y tan bien como podamos, y todo eso. En poco tiempo empezarán a aparecer cosillas para la LPE-Z380, después aparecerán cosas más gordas, y poco a poco empezará a molar. Larga vida al MSX, a las japonesas y al arroz tres delicias.

POSTFACIO

Que en www.zilog.com podeis encontrar toda la información al uso sobre el Z380, así como un ensamblador de idem para guindous 90ytantos (que por desgracia no inluye depurador ni desensamblador). Y que si quereis contactar con el genio de la lámpara, uséase con LP en persona, le encontrareis en lpadial@teleline.es (y a mí mesmo en konamiman@geocities.com, no lo olvideis). No olvideis haceros cargo, y hasta incluso.

Konami Man

```
;Listado 34b: servidor de peticiones de lectura de VRAM, ejecutado por
;el MSX: escanea constantemente una posición de control en la biport,
;cuando encuentra 1 en esa posición significa que el Z380 solicita acceso
;a la VRAM. Entonces lee el dato de la dirección solicitada y lo deposita en
;otra posición de la biport. Finalmente el byte de control es puesto otra vez
;a 0 para que el Z380 reconozca que la petición ha sido cursada.

CONTROL: equ #4300   ;Dir. en biport del byte de control
DATO:    equ #4301   ;Aquí depositará el dato leído, que posteriormente leerá el Z380.
DIR:     equ #4302   ;Dirección VRAM de la que leer el dato

;No hace nada mientras no haya petición
WAIT:    ld     a,(CONTROL)
         or     a
         jr     z,WAIT

;Una vez recibida la petición, la ejecuta y deposita el dato
         ld     hl,(DIR)
         call   SETVRAM ;Rutina que prepara el VDP para lectura desde la dir. HL
         in     a,(#98)
         ld     (DATO),a

;Informamos al Z380 de que la petición ha sido cursada, poniendo CONTROL
;otra vez a 0, y volvemos a empezar (esperar más peticiones)
         xor    a
         ld     (CONTROL),a
         jr     WAIT
```

```
;Listado 34c: petición de lectura de VRAM hecha por el Z380: establece la
;dirección y posteriormente pone el byte de control a 1, después espera a que
;el MSX haya cursado la petición (CONTROL otra vez a 0) y lee el dato.

CONTROL: equ    #40000300   ;Las mismas direcciones del listado 34b,
DATO:    equ    #40000301   ;vistas en el espacio de direccionamiento
DIR:     equ    #40000302   ;del Z380.

;Establecemos la dirección y cursamos la petición (ponemos CONTROL a 1)
         ld     hl,MI_DIRECCION
         ld     (DIR),hl
         ld     a,1
         ld     (CONTROL),a

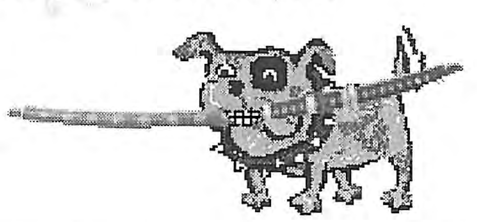
;Esperamos a que el MSX haya cursado la petición, es decir, a que CONTROL ;vuelva a valer 0
WAIT:    ld     a,(CONTROL)
         or     a
         jr     nz,WAIT

;Una vez cursada la petición, recogemos el dato y lo procesamos
         ld     a,(DATO)
         jp     PROCESA
```

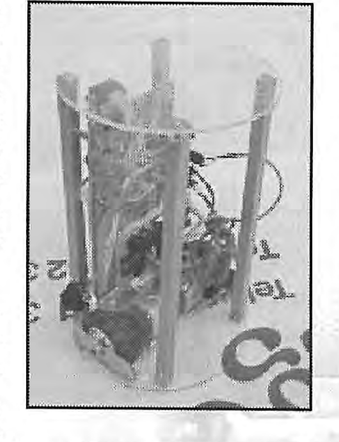
Rigodón. El perro cibernético

No volveré a poner rigodón, no volveré a...

RODON, el robot casero



Hacerse un pequeño robot doméstico es hoy en día barato y muy gratificante. Todo lo que se necesita es unos pocos chips baratos, acceso a un programador de EPROMS, material de construcción de la basura (plástico, metacrilato, madera...), cables, conectores, motores de disketeras e impresoras, ese coche teledirigido que dejó de funcionar... si os pasa como a mí, llegaréis a ver los electrodomésticos como posibles fuentes de componentes electrónicos.



Un robot se puede desglosar en sensores, unidad de proceso y actuadores. En un robot pequeño es posible actualmente integrar todos los chips de medición, decisión y control en un solo chip: un microcontrolador (abreviado, uC). Un uC es un procesador, memoria y varias líneas de E/S en un sólo chip, y está pensado para el control de pequeños sistemas electrónicos. Hay muchas familias de uC's, y en cada familia hay diversos modelos, con diversas capacidades de proceso, memoria y entrada/salida.

Como un uC es un procesador rodeado de pequeños subsistemas, todo en un chip, necesitará un programa que ejecutar. Este programa está generalmente contenido en memoria EPROM, que puede estar dentro del propio chip (y por tanto, cada vez que se quiera programar deberá sacarse del zócalo, meterlo en el borrador UVA, luego en el programador, y de vuelta a la placa) o bien en una EPROM externa. El programa definirá el comportamiento del robot, y programar éste puede ser tan entretenido y gratificante como construir el resto del sistema.

Otros uC's tienen una pequeña EPROM, que se encarga de descargar el programa de un ordenador, vía cable, a una RAM (que a su vez puede estar dentro del uC o en un chip externo) en todos los casos el programa es generado en el ordenador -generalmente un PC, para ser francos-, mediante un compilador o ensamblador cruzado. Se llama cruzado porque el programa generado se ejecutará en una máquina diferente a la que se usa para compilarlo. El compilador cruzado es específico para una familia de uC's, y normalmente es de C, por aquello de que es un lenguaje de alto nivel, pero permite hacer cosas de bajo nivel.

Algunos ejemplos de familias de uC's: 8051 de intel, 6811 de motorola, PIC. El modelo de uC que escojáis dependerá principalmente de la información que consigáis y del compilador que podáis obtener, porque en todas las familias hay prestaciones para todos los gustos. El que yo uso en mi robot Rodón (una

aclaración: el nombre se le ocurrió a un amigo mío una noche de borrachera, y me sabía mal decirle que no me gustaba) es el 8031. La razón es que es el que usamos el Ramoni y yo en una práctica de la universidad, y le empezamos a pillar el gustillo al vicio del silicio. Como tenía información sobre ese uC, posibilidad de usar el programador de EPROMS de la universidad, y compilador, pues seguí usándolo en el robot, por simple inercia. Aunque la verdad es que es un uC muy limitado, como describiré más adelante. La ventaja es su precio: poco más de 200 p.p., y 400 para la EPROM (por comodidad yo me compré una EEPROM, que no necesita "freírse" en el tubo de rayos UVA, con lo que te ahorras 15 minutos en cada programación)

Sobre la familia de motorola que he mencionado, tengo entendido que el 68hc11 es uno bastante bueno. En sus diferentes versiones tiene diferentes capacidades de EPROM y RAM interna, pero generalmente se usa un chip de SRAM externo para el programa, que se descarga desde un PC vía puerto serie. Este uC es de 16 bits y tiene la ventaja de tener muchos puertos, uno de ellos utilizable en modo digital o de adquisición de valores analógicos. En modo analógico puede capturar el valor de voltaje de hasta ocho líneas a la vez, lo cual es muy útil para construir los sensores, y no tener que usar un ADC externo, como tuve que hacer yo. Lo que no sé es el precio del 68hc11, pero debe estar por debajo de las 1000. De la familia de los PIC sólo se que también es muy utilizada y que algunos de ellos tienen sólo un pin de E/S (es una familia muy variada)

EL microcontrolador 8031

El uC que uso tiene las siguientes características:

- Procesador de 8 bits, con direcciones de 16 bits

- 128 bytes de RAM interna (repito: 128 bytes de RAM interna)

- Cuatro puertos de E/S de 8 bits cada uno, P0, P1, P2, P3. Los pins se pueden usar para leer y escribir valores lógicos de otros dispositivos.

- Dos líneas de interrupción externa.

- Dos timers de 16 bits (contadores que generan interrupciones internamente)

- Dos contadores que se activan por el cambio de nivel lógico de dos líneas externas.

- UART y dos líneas (transmisión/recepción que se pueden adaptar al voltaje RS-232 estándar mediante un chip externo.

- Dos líneas de control para usar memoria RAM externa

- Reloj de hasta unos 24 MHz, depende del modelo.

En la **figura 1** podéis ver un diagrama de bloque del micro. No es mi intención Describir los detalles de este chicharro en este artículo, pero sí lo haré en tanto lo necesite para describir cómo he hecho a Rodón. Por internet podréis encontrar mucha información y compiladores, como el Keil 51, que es shareware.

Lo importante es que las interrupciones externas, los contadores, las líneas de conexión serie y las dos de control de memoria externa están en los mismos pines que P3. O sea, que si queréis usar algunas de estas funciones, tendréis menos pines I/O para controlar "cosas" externas.

En la **figura 2** podéis ver otro esquema, con la EPROM y un latch por enmedio, que es una especie de registro. El latch se usa porque en el 8031 la memoria externa se accede multiplexando el byte bajo de la dirección

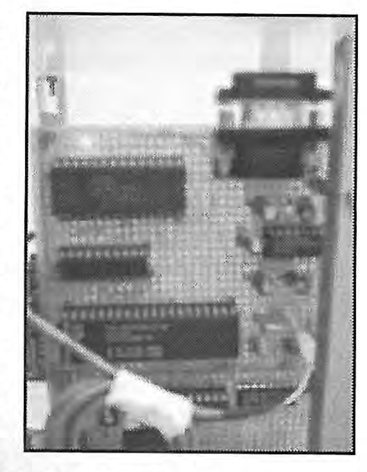
con el byte de datos (leidos o escritos) Esto se hace para ahorrar 8 pins del micro, pero supone poner el latch para mantener la parte baja de la dirección mientras se hace la lectura o escritura. Este latch normalmente es un 74573. Con la entrada LE (latch enable) del latch se controla si se quiere poner el latch en modo transparente (lo que hay en la entrada del latch es lo que hay en la salida), o congelar la salida. El latch es controlado por el micro con el pin ALE (Address Latch Enable). Este sistema de lectura/escritura se utiliza tanto en el fetch de instrucciones desde la EPROM como en el acceso/escritura de RAM externa, si ponemos, o cualquier otro dispositivo que se acceda como memoria. El conversor analógico digital que tiene Rodón es un chip externo y se accede a él de esta forma.

Rigodón

Ha llegado la hora de hablar de Rodón (Rigodón, según el Saver) En realidad empecé haciendo una placa con: - El 8031, el latch 74573 y la EPROM - Un conversor de voltaje MAX232 para conexión serie con el PC - El conversor A/D (ADC) 0801 - Un multiplexor analógico 4051 que sirve para seleccionar una de 8 líneas y leerla con el ADC.

Como dije antes, este ADC está pensado para ser leído como un dispositivo de memoria externo y ser montado en el bus de un microprocesador o microcontrolador (está especialmente pensado para procesadores intel: 8086, 8051, Z80...) Así que el puerto P0 me quedaba libre, y de P3 usaba dos bits para el control de memoria externa, dos para la conexión serie, y otro especial para el ADC.

Quería usar este sistema para hacer un jostick 3D: un brazo de aluminio con potenciómetros en las articulaciones. Los voltajes en los potenciómetros son proporcionales a los angulos de giro, así que enviando los valores digitalizados al PC, podía calcular la posición tridimensional de la punta del brazo por trigonometría. Pero por aquellas cosas de la vida se me ocurrió lo del robot y me puse a pensar cómo usar la misma placa para controlar un cierto número de dispositivos con el puerto libre que me quedaba y los tres bits libres de P3. Lo que hice fue usar P0 como un puerto E/S y usar los tres bits de P3 para seleccionar uno de los dispositivos que conectaría a P0. Decidí poner un decodificador de tres entradas y ocho salidas (un 74138). Lo que hace un decodificador es poner a 0 la salida correspondiente al número binario que hay en la entrada, y el resto de salidas a 1. Así que puedo hacer un nuevo sensor o un actuador, y la circuitería de éste conectarlo a la placa principal con un conector de 9 cables (8 de datos y la señal de activación)



Al principio hablé sobre la capacidad de los microcontroladores para leer sensores, procesar datos y controlar actuadores, todo en un chip. Pero haciéndolo de esta manera consigo construir el robot poco a poco, a medida que se me vayan ocurriendo cosas, o consiga dinero para comprar más piezas.

Puesto que la placa principal ya tenía conexión RS-232, podía usarla para descargar el programa y así no tener que grabar la EEPROM cada vez. Pero en vez de eso, opté por tener el robot siempre conectado con el cable, lo cual tiene dos ventajas: puedo alimentar el robot por cable en vez de usar baterías recargables, y puedo programar el comportamiento del robot en el PC, y mandar comandos de control al micro-controlador constantemente. Por ejemplo, el comportamiento más sencillo es controlar el robot directamente por teclado. La desventaja del cable es el mismo cable, que obstaculiza el movimiento. Pero prefiero eso a comprarme una batería recargable.

Sensores y actuadores

Empezaré por los actuadores. Hasta ahora sólo he hecho una placa que sirve para controlar motores paso a paso, del tipo que se encuentra en disketeras e impresoras. Estos pequeños motores se conectan con cuatro o cinco cables y avanzan en uno u otro sentido un pequeño número de grados. Los motores más comunes dividen la circunferencia en 50 ó 100 pasos, pero los hay de 200 (yo tengo uno). O sea, que una vez conseguidos unos cuantos motores paso a paso, (steppers en inglés), busqué en la red y me bajé un manualillo de cómo hacerlos funcionar. Es muy sencillo, con EL ULN2003, que vale 50 p.p. puedes controlar uno de ellos. Además, estos chips son en realidad 7 interruptores controlados por una señal lógica, así que se pueden usar para encender un foco, un ventilador, un puntero láser o lo que se te ocurra. Así que me hice una placa para controlar seis steppers, pensando en usar dos para el movimiento del robot, dos para una torreta, y dos para ya veríamos qué.

Pero el fallo es que los motores son poco potentes y apenas pueden hacer que el robot arrastre el cable tras de sí. No problemo, cuando consiga dos motores iguales de corriente continua, con engranajes de reducción de velocidad, se los cambiaré. Para controlar motores de c.c. se pueden usar los chips L297D, que pueden controlar el sentido de corriente a través del motor (y por tanto el sentido de giro). Lo malo es que valen 500 p.p. cada L297D.

Con el resto de motores paso a paso que me queden le pondréuna torreta, y al que no le mole que se joda. Nota: son las seis cuarentaiocho de la mañana y vengo de marxa. Estos locos están haciendo y maquetando el fancine en tiempo real desde las 10 del atadre. Yo me he ido, me pegao la maexeta por el gomeielele (Gomila) y vengo to pedo, o wsea que mae vais a escucharere.

Sigo. Eso, que en la totrreta le pondré todos los sensores. Y eso lo explico en el siguiente psarágrafo.

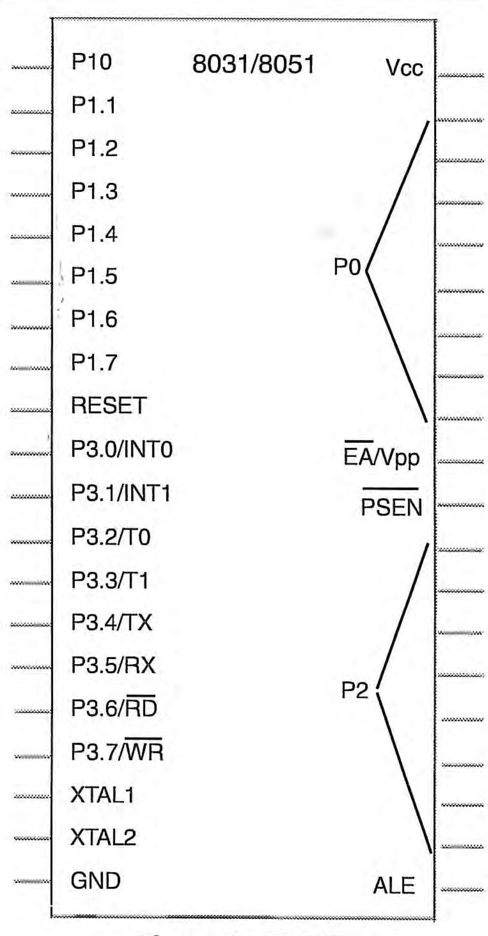
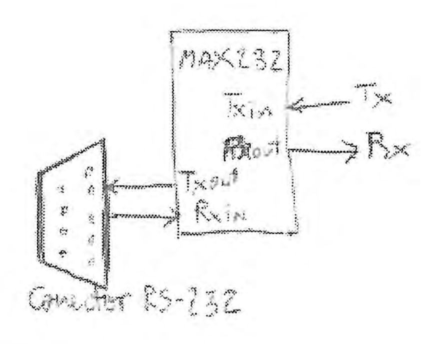


Figura 1. El 8031



Sensores.

Los sensores típicos que se le ponen a un robot son de dos tipos: externos e internos. Los internos son los que miden el propio estado del robot. Ejemplos son tacómetros para medir la distancia recorrida (contando las vueltas que dan las ruedas), sensores de temperatura, etc. Los externos son para medir el medio, como cámaras, sensores infrarrojos de proximidad, bumpers (sensores de contacto por interruptores). Estos son los más sencillos y son los que le pondré primero a rodón (aún no tiene ningún sensor instalado) Después le pondré sensores de proximidad por infrarrojoss, que funcionan con un led infrarrojo y un fotodiodo. En estos sensores se utiliza una señal cuadrada de unos pocos khz, paraeliminar el ruido de las luces, el sol, y electrodomésticos.

Bueno espero que os interese todo este rollo. Pondría algunas páginas web que hablande electronica pero sólo me acuerdo de una (haceros cargo):

http:/ww.robotstore.com

si buscais electronics, o stepper motors, o cualquier otra cosa seguro que encontrais algo. Otras paginas buenas para bajarse datasheets (descripción de componentes electrónicos) es:

htpp//:www.national.com
philips.com
toshiba.com

Apa adeu.

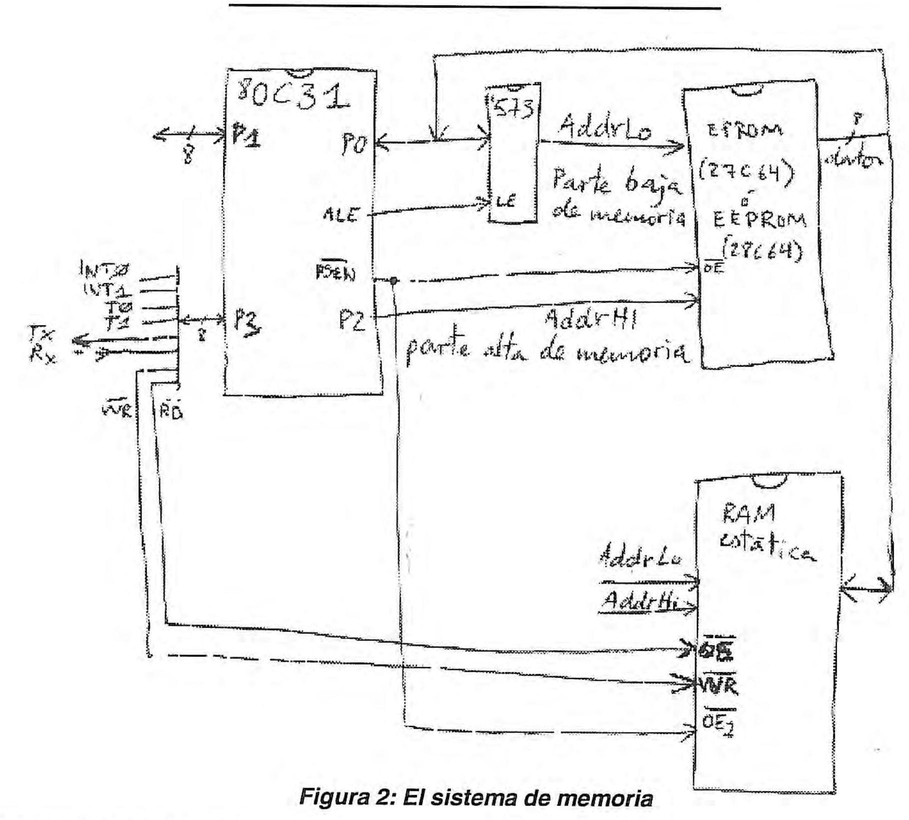


Figura 2: El sistema de memoria

CONCLUYENDONOS

No, no has contado mal. Este número de SD MESXES contiene únicamente la irrisoria cantidad de 48 páginas, como en nuestros arcaicos viejos tiempos. La razón: la de siempre, la falta de tiempo de unos y otros. Hemos preferido sacar el fanzine ya, coincidiendo con la reunión de Barcelona, tal como estaba; la otra opción era esperar hasta el verano, pero un retraso de un año nos parece exagerado incluso a nosotros.

Así pues nos hemos dejado en el tintero muchas cosas: comentarios de reuniones, revistas y juegos; noticias, trucos y anuncios; secciones míticas como el VDP Blaster o el Juocos D'Ace Antaño... incluso la secta se ha visto resentida y reducida a su mínima expresión... por tanto si no has visto tu reunión/revista/juego comentado, o si simplemente piensas que 48 páginas saben demasiado a poco comparadas con las 80 o 90 habituales, y piensas por tanto que este número es una mierda... pues estás en tu derecho. Nosotros sólo podemos pedir disculpas (quedan pedidas), pero por desgracia no podemos prometerte que no volverá a pasar una cosa así, porque la verdad es que ya no damos para más. La vida real está empeñada en robarnos todo el tiempo disponible, dejando a la vida sectaria bastante maltrecha.

- Escenes from de nait of de dekadens -








Y ahora el mensaje pseudo-heorico-patriotero: no importa que cada vez tengamos menos tiempo, no importa que las japonesas nos acosen (si se diera el caso), no importa que el Pajas Maker esté empeñado en batir su récord de problemas por fanzine... en fin, que no importa el asterisco punto asterisco de todo lo que se ponga en contra nuestra: no estamos en absoluto cansados de este vuestro fanzinne (bueno, el ex-capi un poquillo) y en la medida de nuestras posibilidades seguiremos con ello, manque sólo podamos sacar un número cada vez que lo sacamos. Y acabo ya, que se me escapa el avión. No olviden sectarizarse y hacersegarguizarse. Hasta.




FOTO TOMADA DIAS ANTES DE LA DESCAPITANIZACION RAMONERA

En el mundo del Meclese X se habla de cualquier otra cosa menos de la Descapitanización Ramonera. Los tiempos han cambiado y hacerse cargo es cada día más more. Nadie está a salvo, tú puedes ser el siguiente...



KYOKUYO

¿Néstor de capitán? Si no sabe freir un tamago...



CARBONARA

Jeeeeeee eeeeeeeee eeeeeeeeeeeeeee



YIYOLAIS

Eh! Ramoni! pa que te eeechaaas!

NAKAYOSHI

Si yo no le conozco. Dejadme en paaaz!!!!



TOROENTI

Feliz Navidad Ramoni!!



JUANJEN

Ramoniii!!!!! Si eres un hombre!



BRASE

A tu lado soy una pelandrusca!



ZENI-ZERO

Que raro... con lo guapo que es (y el pelo más cuidao)



FROM UP

SaveR, no me provoques...



KIENSOY

Quien quiera que sea.



RIGODON PADRE

Un hombreeee!!!!!!!!



La secta no te olvida... Ramoni

(y ji ji!)

REQUIEM

Ramoni, Ramoni
Todas las viejas
Quieren jugar, con
Ramoni, Ramoni
Que chico más guapo...
¡Que pelo más cuidao!


FOTO TOMADA TRAS LA DESCAPITANIZACION RAMONERA